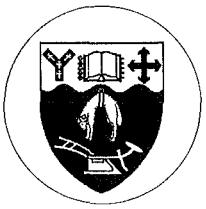
A Meteor Orbit Radar

A Thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy in Physics
in the
University of Canterbury

by

Andrew D. Taylor



University of Canterbury
1991

Abstract

A Meteor Orbit Radar, AMOR, has been built near Christchurch, New Zealand. It uses a narrow beam pulsed radar to detect meteors down to a +12.5 radio-meteor magnitude limit. AMOR measures the relative timelags between the onset of meteor echoes at each of three spaced receiver stations to calculate the meteoroid velocity with an accuracy of ± 2.5 km.s^{-1}. A 'Luggable' MAC AT is used to run the radar, identify meteor echoes and record raw observation records.

A total of 1.3×10^5 meteor velocities have been determined using the new timelag method. For 1.6×10^4 of these observations velocity measurements could be made using Fresnel diffraction patterns and these were in complete agreement with those from the new timelag velocity technique. The diffraction patterns recorded by AMOR allow atmospheric decelerations to be determined by calculating the velocity for overlapping subsections of the patterns. Typically, the atmospheric decelerations found lay between 0 and 40 km.s^{-2}.

Observed meteor velocities need to be corrected for the presence of the Earth before a heliocentric meteoroid orbit can be calculated. AMOR uses a new vector scheme to make corrections for atmospheric deceleration, the rotation and gravitational acceleration of the Earth, conversion to a heliocentric frame using rotation matrices and a correction for the orbital velocity of the Earth. This new derivation greatly simplifies the theoretical framework for computer based calculation of meteor orbits.

The 1990 apparition of the η Aquarid meteor shower was used as a calibration test for the complete AMOR system. 270 shower meteors were detected giving a mean stream orbit with elements of $q = 0.57$AU, $e = 0.98$, $i = 165.5°$, $w = 97°$, $\Omega = 46°$. This agrees closely with previous orbits given for the η Aquarid stream and demonstrates the very large improvement in meteor stream characteristics that can be achieved by AMOR. Orbital elements for individual η Aquarid meteors can be determined within the following measurement limits: $0.33 < q < 0.76$ AU, $0.76 < e < 1.45$, $163.0° < i < 167.5°$, $62° < w < 126°$, $35 < \Omega < 49°$.

Computerised data acquisition, automated reduction to orbital elements and interactive graphical displays were developed and are a significant feature of the system. AMOR is capable of continuous unattended operation producing observation records and reduced orbital elements on site.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Interplanetary Dust and Meteors

A considerable amount of material orbits the Sun between the planets. The larger objects like asteroids and comets can be viewed directly from Earth, while objects smaller than several hundreds of metres across cannot be resolved individually even by powerful telescopes. Dense concentrations of small particles, such as the rings of Saturn, can be seen en masse. Similarly, sunlight reflected from the very small particles, a few tens of microns in size, in orbit around the Sun produces a faint glow which can be seen as the zodiacal light. The presence of objects between these sizes can be inferred when they collide with other bodies. Impact craters on the Moon and micropits on spacecraft are examples of such evidence. More spectacularly when particles hit the atmosphere of the Earth they produce meteors. Very bright ionisation trails, often called fireballs, are associated with a meteorite that survives to hit the Earth. Smaller particles, down to zodiacal dust size, ionise completely in the atmosphere. Both wide angle cameras and television systems have been used to detect faint meteors that cannot be seen by the unaided human eye. Very faint meteor trails can be detected by more powerful radars.

Most of the particles detected as meteors are believed to have come from comets. Each time a comet approaches the Sun, at perihelion, it is heated by solar radiation. The volatile components near the surface are evaporated and ejected from the comet. The solar radiation ionises the gas to a plasma and then blows it away on the solar wind forming a gas tail streaming from the comet. Dust particles are dislodged from the comet as the gas evaporates from the surface. The very smallest, less than one micron, of these are also accelerated by the solar wind and form a distinctive dust tail or fan. Larger particles remain in much the same orbit as the parent comet. The small range of ejection velocities from the nucleus combined with subsequent gravitational perturbations by the planets spread these larger dust particles around the comet orbit forming an associated meteoroid stream. Close plane-

tary encounters move some of these stream particles into random or sporadic orbits. Various radiation effects change the particle orbits more slowly. Interplanetary collisions between particles shatter the dust producing a range of fragment sizes. Remnants smaller than about one micron are accelerated by the solar wind and blown out of the Solar System.

As the Earth moves around its orbit a tiny fraction of this interplanetary dust complex will hit the atmosphere. By observing the ionisation trail as it forms behind a meteor it is possible to measure the particle velocity. From this, the heliocentric meteoroid velocity can be calculated and hence the orbit of the interplanetary dust particle. Observations of meteor orbits can be used as a probe for the distribution and dynamical characteristics of interplanetary dust.

An orbit radar has been built at Christchurch, New Zealand to observe meteors and calculate the associated meteoroid orbits. This installation, A Meteor Orbit Radar (AMOR), is the subject of this thesis.

The radar is expected to give a greatly increased volume of orbit data on major meteors streams. This will provide a greater statistical base on which to base mean orbits for the streams. For example the mean orbit given for the η Aquarid stream in the list of streams by Cook *et al* (1972) is based on one observation. The order of magnitude increase in data should make it possible to investigate fine structure in these major streams.

The AMOR system is capable of detecting meteors to a radar magnitude limit of +12.5 to +13 thereby observing particles down to a size of about 100 μm. The distribution of the orbital element for sporadic meteors in this size range is of interest in understanding its potential as a reservoir of material to form the zodiacal dust cloud.

Radar reflections from meteors can be used as a probe to study the structure and motion of the atmosphere. The diffusion and body motion of the ionisation trail in the neutral atmosphere allows for making measurements of density and wind velocities to be made. Knowledge of the orientation and track of the particle forming the ionisation trail provides useful extra information in studies of this kind.

Over 100 asteroids that cross inside the orbit of the Earth have been discovered. This group of minor planets is divided into three classes named after the first such object discovered. The Apollo asteroids cross inside the orbit of the Earth with a perihelion distance of less than 1.017 AU. Aten asteroids form a subset of this group with an orbital semi-major axis of less than 1.0 AU. Amor asteroids have a perihelion distance of less than 1.3 AU but do not cross inside the orbit of the Earth. There are too many asteroids in this Apollo-Aten-Amor complex for the group to be dynamically stable. Close encounters with planets would cause them to be ejected from the inner Solar System in times that are short compared with the age of the Solar System.

There is considerable debate as to the origin of these asteroids. Ejecting objects from the asteroid belt into Earth crossing orbits is dynamically diffi-

cult. Deactivated comets have been suggested as a candidate source. If these asteroids were once comets they are likely to have meteoroid streams associated with them. For those asteroid orbits which intersect with the orbit of the Earth these cometary fingerprints should be evident amongst the meteor orbit distribution. One of the primary motivations for developing the Christchurch radar was to search for meteor streams associated with near Earth asteroids. The AMOR acronym is doubly appropriate.[1]

1.2 Visual Meteor Observations

In the later part of the 19^{th} century the importance of acquiring orbital information on meteors was recognised by Schiaparelli and Newton. Some indication of meteor radiant and velocity distributions of the meteoroid population can be achieved from observations of the diurnal, seasonal and latitudinal variation of meteor influx. A quantitative model fitting approach along these lines was first considered by Schiaparelli (1866). It was later used by Hoffmeister (1948) for example to delineate crude orbital discriminations thought erroneously to have a large interstellar component.

1.2.1 Visual Orbit Determinations

Denning (1899) and Prentice (1945) in Britain used angular velocity estimates of visual meteors in attempting to determine whether meteoroids were travelling in closed heliocentric orbits. Meteor trajectories were mapped using coordinates superimposed by the observer and angular velocities calculated by timing estimates.

The first precise quantitative information came from the rocking mirror device of Shapley *et al* (1932). Meteor tracks were measured using a fixed reticle of hour angle and declination with angular velocity estimated by the rocking mirror. An observer watched the reflected meteor track. The pattern of loops together with height data gained from simultaneous observations at stations separated by a 40 km baseline gave velocities. The field of view was 60° and meteors down to +4 magnitude could be recorded. For fainter meteors the rocking mirror could be used in conjunction with 4 inch reflecting telescopes. The main emphasis of this work carried out during the Harvard-Cornell Arizona Meteor Expedition 1931-3 was in securing geocentric velocities. This visual survey concluded that about 60% of sporadic meteors had velocities exceeding the parabolic limit (Öpik, 1948).[2]

[1] Of course any PhD thesis turns into a work of love at some stage and AMOR seemed appropriate in this sense as well. I discovered the word while at the First Vatican Summer School (1986) in Astronomy and Observational Astrophysics, hence the Latin or Italian spelling.

[2] This conclusion was later criticised by Lovell (1952). The length of the baselines used and problems with calibration were the principal causes of doubt.

1.2.2 Photography

More objective records started with the application of photographic techniques to meteor observations. Some of the earliest meteor photographs were obtained on the Harvard photographic patrol programme in the 1890s.

The use of separated cameras with occulting shutters over their lenses to give angular velocity was first employed by Elkin (1899) at Yale in the U.S.A. 1894-1909. The system baseline of only 3.5km limited the resolution obtained on the 131 photographic tracks later analysed by Olivier (1937). Improvements in emulsions and cameras allowed for a long programme at Harvard initiated by Fisher and Olmsted (1929) and continued by Millman. From 1936 the programme continued under Whipple who was able to derive atmospheric density values from measured meteor decelerations.

The cameras employed in these surveys were only sensitive to meteors brighter than magnitude -1. With their angular field of 60° the acquisition rate was only 1 per 100 observing hours. This limitation of only a handful of orbits over the years of operation was overcome with the Super Schmidt cameras (limiting magnitude +4) operated by Harvard University at Soledad Canyon and Dona Ana in New Mexico. 2800 orbits were secured in the period 1952-9.

Other multi-station camera determinations of orbits has been carried out at Dushanbe, 1975-83; Ondrejov Observatory (Czechoslovakia) 1951-86; Odessa and Kiev. Much of the published Super Schmidt orbit results appear in the Smithsonian Contributions to Astrophysics series of publications (for example Vol **4** no 2 and no 4, 1961). More recently photographic observations of meteor orbits have been continued by amateur groups using 35mm rotating shutter spaced cameras in Netherlands (Beltlam and de Lignie, 1989) and in Japan (Koseki, 1989).

Orbits of fireballs have been secured by fisheye camera networks. The Prairie network in the U.S.A. under the direction of the Smithsonian Institute operated in the period 1963-75. The European Network of Central Europe with 52 stations (as of 1990) in Czechoslovakia, Germany and Netherlands has continued to produce orbits from 1964 to the present. The Meteorite Observation and Recovery Project (MORP) has operated in Canada from 1971. Meteor orbits from MORP are presented by Halliday *et al* (1983) and (1989).

1.2.3 TV Cameras

With the advent of sensitive TV photo receptors, recording of trajectories has been possible for meteors fainter than the Super Schmidt limit. Hawkes *et al* (1984) using image intensifiers in conjunction with vidicon cameras achieved magnitude +3. Later with ISIT cameras Jones and Sarma (1985) reached a limiting magnitude of +8.5. Individual frames of camera video tapes from sites separated by 30km baselines were examined with a digitiser to achieve 6 arc

sec angular accuracy. This translates to an accuracy in the orbital elements of ± 1-2°, with the eccentricity accurate to about ±0.05.

1.3 Radar Orbit Measurement Techniques

Some information on the radiant distribution of meteors can be gained by making use of the specular reflection property of the radio scattering process from an ionised column. A statistical approach using occurrence-time and radar range-time characteristics obtained using a radar with antennas of known angular response was proposed by Clegg (1948). This was used at Jodrell Bank to isolate shower radiants and to deduce the sporadic radiant distribution over the celestial sphere. A combination of radiant coordinates obtained from this statistical method together with velocity information (section 1.3.2) permitted the determination of a representative shower orbit.

To determine the heliocentric orbits of individual meteors it is necessary to locate the radiant and hence the direction of the meteor trajectory and the pre-atmospheric velocity for the particle. The actual position of the ionised column in space with respect to the Earth is not actually required. Essentially three receiving stations as first proposed by Kaiser (Hawkins, 1964) are required.

Three principal methods have been employed for the determination of individual meteor orbits.

1.3.1 Range-Time

The reflection process of the radar echo from the main body of the ionisation trail is one of specular geometry. The echo strength is due to the contributions of several Fresnel zones on either side of the specular point. However, for very bright meteors the body echo is often accompanied by a head echo. This seems to be associated with a ball of ionisation travelling with the meteoroid in front of the developing ionisation column (Simek, 1973). A range-time plot of the reflections from this plasma ball will describe a hyperbola from which the atmospheric velocity of the particle can be measured. The first such determination was by Hey and Stewart (1947) in their measurement of velocities for the Giacobinids during the 1947 apparition. Three station range-time comparisons yielded the meteor location and trajectory. This method can be used only for very bright meteors producing intense ionisation.

1.3.2 Pulsed Radar

During the formation of the meteor trail, diffraction effects analogous to Fresnel straight edge diffraction are responsible for an oscillatory scattered signal. When the meteor passes the strict orthogonality position (specular reflection point) half the zones contribute in phase to the reflected echo strength resulting in a half amplitude echo. As the meteor continues to move past this point

each Fresnel zone successively adds in and out of phase thereby adding and subtracting to the amplitude of the signal. The period of these oscillations in the diffraction pattern permits the atmospheric velocity of the meteor to be determined. Modifications to the ideal straight edge function due to diffusion, meteor deceleration and wind shear have been considered, for example by Southworth (1972). The first measurement of a meteor velocity using this technique was reported by Ellyett and Davies (1948). Employing three spaced stations will yield three diffraction patterns. A separation of about 5 km between the receiver sites ensures that the reflection points on the meteor trail are separated by a small distance compared with the overall length of the ionised column. Timing differences in the occurrence of the echo profile at each site each gives the direction of the trajectory. Ranges to the meteor trail are measured by pulse timing.

A coherent phase system can be added to the pulsed radar. By providing a reference signal to the receiving sites, Doppler measurements of the atmospheric wind can be made and then wind effects allowed for in data reduction. The presence of a reference phase makes it possible to measure oscillations in the diffraction pattern as the meteoroid approaches the specular reflection point. The pattern continues symmetrically after the point of closest approach.

1.3.3 Continuous Wave Technique

A CW radar measures the differential phase between the reflected signal and the transmitter ground plane reference. The transmitter and receiving stations need to be separated by some tens of kilometres to reduce interference. This phase information is used to calculate the direction of arrival at each of several, but a minimum of three, receiving sites. Direction cosines to the meteor from each receiver site can be determined. Additionally the coherent phase CW operation permits measurement of the fluctuations due to Fresnel zones formed both before and after the meteor particles motion through the specular reflection point. This has the advantage of being less affected by trail distortion and diffusion effects.

1.4 Review of Previous Orbit Radars

McKinley and Millman (1949) extended the range-time dependence of head echoes to three stations using omni-directional antennas to determine a few orbits. Being limited to very large meteors the successful orbit rate was only about 1 per 100 hours of observing. Manning *et al* (1948) and McKinley (1961) used CW systems with photographic recording of a single frame triggered camera.

Gill and Davies (1956) describe an incoherent 36.3 MHz pulsed radar system used at Jodrell Bank. Davies and Gill (1960) note that the transmitter station produced 100 kW peak power with a pulse frequency of 600 Hz. A

Location	Dates of Operation	Number of Orbits	Limiting magnitude	Reference
Ottawa	1948	-		McKinley and Millman (1949)
Stanford	1948	-		Manning *et al* (1948)
Ottawa	1950	-		McKinley (1961)
Jodrell Bank	1954 - 1955	2509	+8	Davies and Gill (1960)
Khankov	1960 - 1966	12,500	+7	Lebedinec (1968)
Adelaide	1961	2092	+6	Nilsson (1964)
Havana	1961 - 1969	40,000	+12	Cook *et al* (1972)
Kasan	1968	1200	+8	Andrianov *et al* (1970)
Adelaide	1968 - 1969	1667	+8.5	Gartrell and Elford (1975)

Table 1.1: Review of Previous Orbit Radars

transmitter antenna beam 24° wide with a gain factor of 28 was used in conjunction with spaced receiver stations with aerial gains of 12 which were separated by 3.5 km baselines. Photography of amplitude-time displays recorded the data and orbit determinations were attempted on selected echoes showing at least three diffraction zones.

Nilsson (1964) ran a combined CW and pulsed system with four outstations. The transmitter was separated by 20 km to reduce the amplitude of the ground pulse. The transmitter operated on 27MHz and produced 10kW pulse with 300 watts CW. The aerial system used a vertical 3 element Yagi for the transmitter and half wave dipoles a quarter wavelength above ground for the four receiver stations. The resultant power distribution was omnidirectional in azimuth and broad in elevation with a maximum to the zenith. The system produced on average 1.3 orbits per hour of observing. Reduction of signals was accomplished by photography of displays, filmreader to punched tape and finally to punch cards. The update reported by Gartrell and Elford (1975) employed a 65 kW pulse at 200 Hz on 27.5 MHz, with 26.8 MHz 1.5 kW CW and six outstations.

In terms of sheer RF power the pulsed system reached its zenith with the Havana radar described by Cook *et al* (1972). It had a design limiting magnitude of +12 achieved by a 2 MW transmitter 738 Hz pulse rate, with every 5^{th} pulse doubled or delayed for ranging. A test transmitter towed by an aircraft was used to calibrate the antennas (gain 22dB). From 1963-69 a phase coherent system was installed with a 2.5 kHz reference signal provided by telephone lines and later microwave links. With six outstations many reflection points were available for wind profile measurements and used to improve the diffraction pattern analysis. The system was so overwhelmed by the volumen of data recorded that it was only run for five minutes in every hour of operation.

The use of Fresnel diffraction patterns for velocity measurements employed by previous orbit radar systems suffers from two disadvantages. To measure

well defined echo amplitude fluctuations for fast meteors a high pulse frequency of greater than approximately 600Hz is necessary thus introducing aliasing in range measurement. Fragmentation produces multiple sources damping out the resultant Fresnel pattern while wind distorts the profile. Davies and Gill state that only 10% of echoes in their survey were useable. In the present AMOR radar we find approximately 4% yield simultaneous diffraction patterns from 2 or 3 receiver stations while approximately 10% give a pattern on one. Routine velocity measurement in AMOR does not use diffraction patterns but rather times the intervals between the appearance of the echo profiles on spaced receivers. However Fresnel velocities do provide an important calibration.

Orbit determinations of very faint meteors have been achieved in general by substantially increasing the transmitter power. Building a 2 MW transmitter similar to that used in the Havana radar is a very difficult engineering exercise. The same increase in magnitude can be achieved by increasing the antenna gain with some sacrifice in angular coverage at any one time. This approach is used by the Christchurch AMOR system to reach a limiting magnitude of +12.5 with a 20 kW peak power pulsed transmitter.

Previous orbit radars have used photography of oscilloscope displays to record the data. The film was developed, measured and reduced thus requiring a considerable labour in selecting and measuring appropriate observation records. More recently computers have been used to speed the calculation of orbits from this measured data. The Christchurch radar was conceived as a microprocessor based system with receiver output voltages converted directly to a digital amplitude. Several chains of microprocessor boards were designed and built giving a multichannel system that recorded radar information in a machine readable form for subsequent computer analysis.

1.5 Christchurch Meteor Orbit Radar

A meteor orbit radar (AMOR) has been developed at the Birdlings Flat field station of the University of Canterbury, Christchurch, New Zealand. Three major innovations distinguish this radar system from previous installations. The system uses narrow antenna beams which implicitly locate the meteor and increase the radar gain making possible the detection of fainter trails. AMOR calculates the timelags between the occurrence of meteor echo profiles on three spaced receivers to determine the particle velocity. This gives a considerable increase in the number of meteor observations for which velocities can be determined compared with previous Fresnel diffraction methods. Finally, computerised data acquisition, storage, reduction and display make it possible to completely reduce observation data within 24 hours of its collection. The association of data acquisition and reduction software with diagnostic display routines provides a powerful environment in which to maintain the system and conduct any further development. A graphical display suite provides ready ac-

cess to the AMOR data base.

A meteor orbit radar needs to locate the meteor in space and determine its velocity. Effectively six kinematic degrees of freedom need to be determined for the particle. As the meteoric particle is too small to be seen directly this is done by watching the associated ionisation trail formed in its wake. The AMOR system uses a 26.2 MHz pulsed radio transmitter to detect meteoric ionisation trails. By timing the delay between the sending of a pulse and the detection of the meteor echo the range to the target trail can be determined thus fixing one of the positional degrees of freedom. Radar pulses are sent by the transmitter every 2.64 ms making it possible to determine unambiguous ranges to meteor trails out to about 400 km.

The Christchurch AMOR radar uses long broadside antenna arrays which concentrate the antenna gain in a narrow azimuthal beam with a 3° width. These beams are aligned to point north-south and hence locate the azimuth angle to the meteor trail. A rhombic transmitter array directs most of the radar power south making detection of meteors travelling from north of the station more likely. The antennas are suspended about 0.6λ above the ground and produce a broad beam in elevation. Two spaced antennas at the Home site of the radar station act as an interferometer to determine the elevation angle of the returning echo. The range, elevation angle and implied azimuthal angle locate the meteor trail relative to the radar station.

AMOR uses a backscatter configuration to detect the meteor trails. The ionisation column formed by the meteor acts as a long thin reflector at the 11.45 m radar wavelength. To receive an echo at the radar station the trail must be at right angles to the radar beam. The particle must be travelling in a plane normal to the direction of the received meteor echo. As the meteor moves past the specular reflection point the echo amplitude increases as the ionisation trail forms. The column then diffuses into the neutral atmosphere and the echo strength decays away. The echo amplitude associated with each transmitter pulse is recorded by the system giving an amplitude profile with a 379 Hz sample rate. By comparing the timelags between the rising edge sections of the echo profiles from three spaced receiver stations the final two components of the velocity can be determined. The two remote receiver sites are separated by 8.18 km and 10.54 km from the Home site.

The current philosophy behind the equipment design is to digitise the radar data and insert it directly into the computer memory of an IBM PC-AT compatible as quickly as possible. Detection algorithms written in high level software are initiated by an interrupt every 2.64 ms to locate meteor echoes in the previous radar range sweep. From here the observation data necessary to calculate a meteor orbit can be extracted and stored. The time of occurrence, range, duration of the trail, three echo amplitude profiles and two phase angle profiles are recorded for each meteor observation. From this information the relative position and velocity of the meteor particle are calculated.

Before calculating the meteoroid orbit a series of corrections need to be

made and the position and velocity converted to a heliocentric reference frame. The orbit reduction package employed by AMOR to do this uses a new technique based on vector notation for the meteor velocity. From this the orbital elements for the meteoroid's motion around the Sun are calculated.

1.6 AMOR Thesis Layout

This thesis discusses the meteor orbit radar developed at the University of Canterbury, Christchurch, New Zealand. It describes the radar hardware, observing software and data reduction techniques. A new method for converting the observation data to a heliocentric orbit is developed. The Fresnel diffraction method for determining meteor velocities is investigated as a check on the timelag based method used by AMOR. The 1990 observations of the η Aquarid meteor shower are used as an astronomical calibration of the radar.

The chapter on the radar hardware describes the main features of the radar installation. The performance characteristics of the 26.2 MHz radio frequency components are covered in addition to the phase comparison system of the Home site interferometer. The design and theoretical gain calculations for both the the transmitter and receiver aerial arrays give an estimate of the gain achieved. The work involved in checking the radiated power distributions and alignments for the arrays is highlighted. A limiting radar magnitude of +12.5 is found for the AMOR system. The operation of the two FM data links from the remote receiver sites is described. The digital hardware which controls the radar and the operation of the direct memory accessing to get meteor echo data into the computer are both covered in Section 2.5. A brief historical note on equipment superceeded in the period 1978–91 is included.

The observation software runs the computer during an observing run. The associated data structures and memory use within the program are described. Every 2.64 ms an interrupt is sent from the control hardware and initiates the execution of a meteor detection routine. The assembly of the observation data record necessary for an orbit calculation is described. Finally in this chapter the temporary storage of observation records in extended memory and subsequent archiving is covered.

Data Reduction for the AMOR system requires the position and velocity of the meteor particle to be calculated. The position of the meteor is discussed in sections on the range to the trail, elevation angle of the echo with its associated calibration and the resolution of ambiguous phase angles. Several methods for determining the receiver station timelags and hence the meteor velocity are discussed in detail. The method based on the location of the maximum rising edge slope was chosen for the AMOR system. The calculation of the maximum echo amplitude includes a discussion of the techniques used to reduce the impact of noise and interference on data reduction. The relation of the recorded echo amplitude to the RF voltage detected by the receiver is

also discussed. Estimates of the meteor diffusion heights are used as a check on the calibration of the elevation angles measurements. Finally the velocity vector of the meteor is calculated in terms of its azimuth angle, zenith angle and atmospheric speed.

The reduction and analysis of Fresnel diffraction patterns in the echo amplitude profiles was investigated as an independent check on the velocity determination scheme used by the AMOR system. The calculation of an estimated mean echo amplitude and the automated analysis of oscillations about this mean profile are discussed. Combining Fresnel velocity determinations from several stations allows an estimate of atmospheric deceleration to be made on a small fraction of the observed meteors. A comparison with the timelag velocity determinations gives a good agreement including identification of several meteors from definitely hyperbolic meteor orbits.

The chapter on the calculation of the orbital elements gives a detailed derivation of this process. It describes a new approach using vector notation and matrix rotation to handle the necessary corrections and coordinate transformations. A correction is made for deceleration in the atmosphere prior to detection. The relative velocity due to the rotation of the Earth is subtracted. The corrections for both the increase in speed and change in direction (zenith attraction) due to acceleration in the gravitational field of the Earth are discussed. The transformations required to convert from the station frame first to equatorial coordinates and then to the ecliptic heliocentric system are itemised. The orbital motion of the Earth is subtracted giving the final heliocentric velocity of the meteoroid. Assuming this to correspond to the position of the Earth at the time of observation, the orbital elements of the meteor are calculated. General equations of motion using the notation of this thesis are derived for a particle moving in a central gravitational potential. A worked example for this calculation scheme is given for a typical η Aquarid meteor observation.

The η Aquarid meteor shower was used as an astronomical test case for the AMOR system. Chapter 7 describes the 1990 observations of this shower. An initial selection of 361 shower meteors was done based on the corrected geocentric radiant position and magnitude of the heliocentric velocity. These showed a clear daily motion of the radiant indicating the motion of the Earth through a meteoroid stream. The measured distribution of orbital elements for the selected stream members is presented. An analysis of the measurement uncertainties associated with η Aquarid observations by the AMOR system shows a broadly similar set of distributions. The dispersion in individual elements is probably due to measurement uncertainties, thereby justifying the uncertainties attributed to the measured station data. The distribution of orbital elements for the stream contrasts markedly with the general background detected during the observing run. An alternate method of selecting stream meteors based on their calculated orbital elements and using the modified Drummond D-Criteria is described. A direct search based on the mean

η Aquarid orbit and a serial search seeded from the same orbit are both tried. The serial search gives 270 η Aquarid stream members for the 1990 observations. A series of meteoroid orbital density cross sections are presented to confirm that these stream associations are the product of a real concentration of meteor orbits rather than a product of the selection criteria used to choose the stream. A more general search is also conducted based on the present orbit of Comet Halley to show that a meteor stream is associated with that comet. Some evidence of fine structure in the stream is investigated. The mean orbital elements for the η Aquarid stream agree well with previous measurements. The number of 1990 observations represent an order of magnitude increase over previous orbit determinations for the η Aquarid shower.

The final chapter summarises the operational station layout of the AMOR system. Uncertainties associated with the radar measurements and the basic stations parameters are tabulated. A glimpse of the graphical display options for viewing the fully digitised orbital data set is given. Suggestions of future developments for the AMOR system are combined with a discussion of the inherent limitations of the system.

Three main appendices provide a reference manual for the computer software associated with the AMOR project. These are all written in Turbo Pascal 5.0. Appendix A gives brief descriptions for about 120 programs developed on the PC during the last two years and considered important for a maintenance and development base for AMOR. Turbo Pascal uses a system of code modules to break software up into manageable files. Descriptions of the routines available within these subroutine library units are contained in Appendix B. Listings of the computer code directly involved in servicing the radar during an observing run and subsequent data reduction are included in Appendix C. A final Appendix D showing the overall distributions of measured station data and orbital elements is included. This includes data for the approximately 1.32×10^5 meteor orbit observations made in the 14 months since the AMOR system began routine operations.

Chapter 2

Radar Hardware

The physical components of the AMOR system are described in this chapter. Successful operation of the radar requires the reliable simultaneous functioning of all these components. The process of achieving this was a substantial part of my thesis work. Individual components were made more reliable or modified to work as required. Occasionally this involved redesign or even complete abandonment of the component. All the components described here, except the transmitter, have been modified, built or rebuilt at least once as part of my research work. The hardware falls into several groups each discussed in a separate section.

The computer is a IBM-compatible MAC AT. The system was assembled in a 'Luggable' case rather reminiscent of a 'portable' sewing machine. It has 640 kBytes of directly accessible memory and a 385 kByte extended memory. The computer operates on a 12 MHz clock cycle[1]. A 40 MByte hard disk drive is installed for data storage and program development. Software packages and

[1]It was found necessary to upgrade the RAM from 120 ns to 100 ns chips to make this possible. The observing software was developed to operate at the 12 MHz speed. In trying

utilities are kept to a minimum. An Ethernet card provides communication facilities to the rest of the world.

In the spirit of a first year laboratory report I have made a list of the major components to give a feel for the complexity of the system and the sheer logistical exercise of getting it all working:

Transmitter	1	A/D Conversion channels	5
Transmitter Array	1	Timing Control Board	1
Receiver Aerial Arrays	4	DMA Interface Card	1
26.2 MHz AM Receivers	4	Luggable AT Computer	1
RX Phase Comparator	1	VAX Mainframe	1
FM Link Transmitters	2	Magnetic Tape Drive	1
FM Receivers	2		

2.1 Radio Frequency Components

The AMOR system detects meteors using a radar operating on 26.2 MHz. A trigger pulse is sent to the transmitter by the digital control board. The transmitter produces a 66 μs pulse. This is amplified to deliver a 7kV, 70×40 mA pulse to the transmitter aerial array. The returning echo is collected by the receiver arrays. The subsequent passage of the signal through to the A/D converters is outlined in Figure 2.1. At the remote sites the video signal is fed directly into the FM link transmitter. At the Home site the FM link receivers demodulate the signal and send the amplitude information to the A/D converters. The signal from the south and north Home site aerials are each fed into a 26.2 MHz receiver. The video signal from the south receiver gives the Home site echo amplitudes. The local oscillator in the north receiver provides a reference for the phase linked south receiver. The video outputs from the two linked Home site receivers are fed into the phase comparison hardware. This produces a linearised sine and cosine amplitude output that is converted to a digital signal.

2.1.1 Transmitter

The 26.2 MHz transmitter was built in 1969. Since then it has had a long history of service. The transmitter was designed for an average power input of 1 kW. A maximum average power output of around 700 Watts could be expected. The transmitter is run at 7 kV and 70 mA. This gives an average power output of 490 Watts. Running the transmitter well under full power provides a considerable enhancement in reliability. Doubling the power would

to diagnose an intermittent memory fault a piece of conducting foam was removed from the CPU bus connections. The computer would then only run at the alternative 6 MHz clock rate. This caused some radar pulses to be missed. The memory was upgraded, 12 MHz operation restored and we have not had a problem with the memory since.

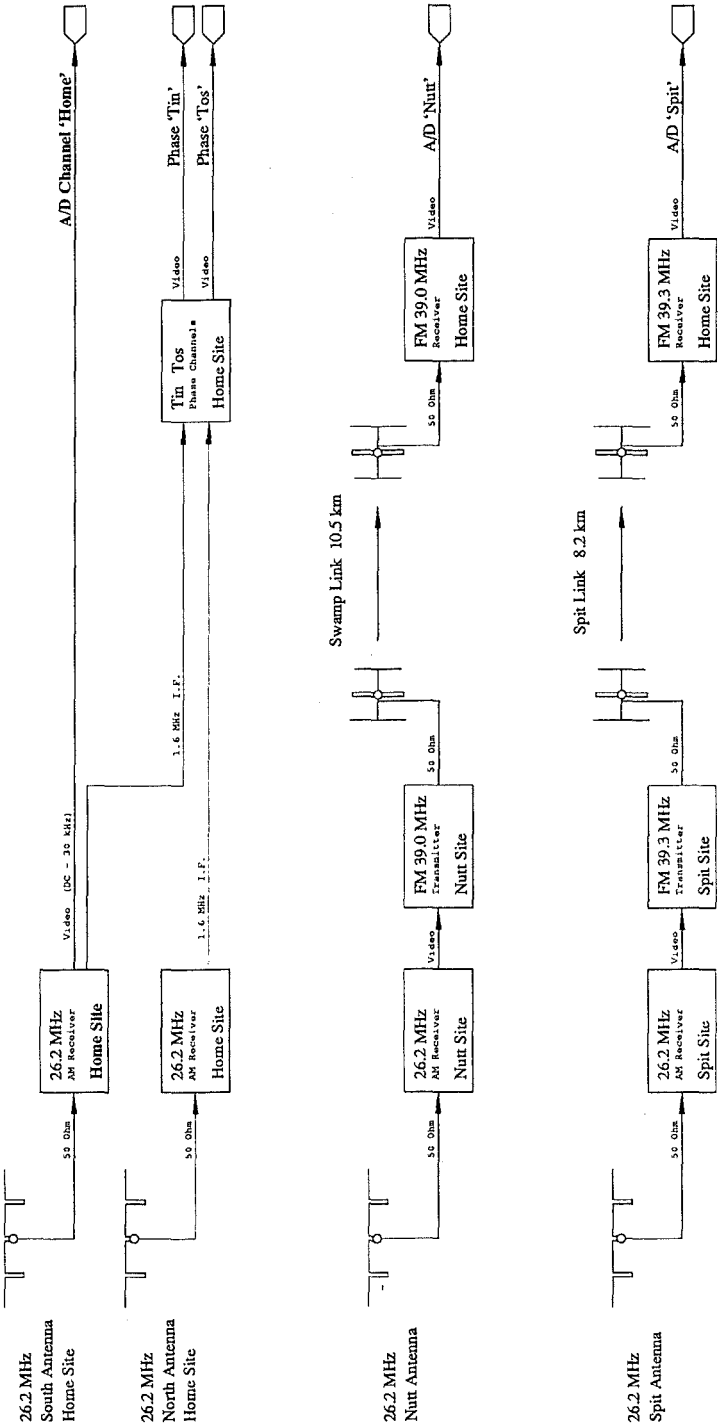


Figure 2.1: The radio frequency layout of the AMOR system. The schematic shows the data flow from the 26.2 MHz aerials through to the A/D converters and digital data logging system.

only give a $\sqrt{2}$ improvement in the echo strength. The extra labour involved in maintaining this is not justified.[2]

With its advanced age the number of component failures in the transmitter has increased. It seems that power surges through the mains produce a fair number of these. A combination line filter and power regulator was added, substantially reducing the number of failures. This unit was contained in an orange box. The advantageous seemed to be confirmed with the increased failure rate when the original orange box was stolen after several months of trouble free operation. The meteor transmitter shares the transmitter hall with a 2.4 MHz transmitter used to measure mesospheric winds by partial reflection drifts. The concurrent operation of these transmitters produces a heavy load with considerable voltage drop on the mains transmission to the building. The presence of the mains regulator helps to keep the power of the the meteor transmitter up.

The radar pulses should be longer than the microsecond that it takes the transmitter output to rise from zero to peak power. With increased pulse length the range resolution deteriorates. The transmitter produces a relatively long 66 μs pulse to get power into a narrow bandwidth.

A trigger pulse is sent to the transmitter every 2.64 ms. This produces a pulse rate of 379 Hz. This pulse rate combined with the pulse duration produces a duty cycle of 40:1. The 490 W average power is concentrated into 20 kW pulses. Within the limits of average power noted above the transmitter could be used at any pulse rate. Originally the power supply of the transmitter was composed of three 1μF capacitors. The continuous operation of the transmitter at this high pulse rate increased the duty cycle on these capacitors. The dielectric started to melt and ooze out the top of them. They were replaced in 1988 with two more modern 2 μF capacitors half the physical size of each of the originals. The power supply is capable of producing 1.6 kW DC power.

The trigger pulse is amplified through four stages and used to trigger the Driver. The final amplification is done by four C1149/1 tetrodes in the Power Amplifier[3]. An adjustable inductive coupling connects the final stage to the TX transmission line. [4]

[2]The time lost whilst the transmitter is repaired reduces the number of meteor orbits obtained. This would probably eliminate the increase in orbit observations gained by increasing the power.

[3]Having done some research work on lung physiology I always thought the PA was some oblique reference to the Pulmonary Artery.

[4]When I originally met the transmitter I thought of it as a large black box that produced 66 μs radio pulses at the prescribed rate. This acquaintance has now been refined to a stack of black boxes plugged into an orange box that does exactly the same thing.

2.1.2 Receivers

The AMOR system contains four 26.2 MHz receivers. A circuit diagram describing the electronic layout of them is found in Figure 2.2. The receivers have an input impedance of 50Ω. A tuned circuit and amplification stage precede the mixer in the receiver. A 24.6 MHz crystal provides the local oscillator reference. Two 23.0 MHz traps are included to filter out the IF image. Three stages tuned to the 1.6 MHz intermediate frequency amplify the signal on the IF strip. Finally the video receiver output is generated. This ranges from 0 to 10 volts. The signal is taken along the backplane of the receiver rack and then through a coaxial cable to the A/D converters. The noise at the output of the receivers, without a 50 Ω load attached, is equivalent to 0.2 μV at the input.

The individual receiver bandpass characteristics and amplitude calibration curves are discussed below. In general they have a 20 kHz bandpass. Within the receiver power range the video output is reasonably linear. For the Home site receivers this corresponds to a signal strength ranging from 2 to 35 μV. The video output on the remote site 26.2 MHz receivers is set low to ensure the modulated FM signal of the links stay within their frequency range. The final video voltage to the A/D converters for the remote site channels is controlled in the FM link receivers. The calibration curves for the remote site channels show the amplitude response after transmission through the FM links.

Receiver Bandpass

The receiver bandpass diagrams for each of the 26.2 MHz receivers are shown in Figure 2.3. The bandpass curves of the remote site channels as output by the FM link receivers are also plotted. A Marconi 2019A digital signal generator was used to measure these curves. Using a constant input signal strength the generator was scanned through the frequency range in 1 kHz steps. The video level for each point was measured using the A/D converters and then recorded by the computer. The data can be collected with the BandRX.Pas program and displayed with BandDraw.Pas.

The receivers were designed to have a 3 dB bandwidth of 20 kHz. The measured bandwidths range from 20 to 25 kHz (as may be seen in Figure 2.3) and the response curves reasonably symmetric about 26.2 MHz. The combination of the Home site receiver without the local oscillator and the south aerial seemed to be least susceptible to interference from external sources.

Using the signal generator as described above is a rather cumbersome way of measuring the bandpass characteristics. It does exercise the whole system and provide a digital record of the receiver bandpass curves. A much more rapid check can be obtained by using the ramp pulse from an oscilloscope to FM modulate the generator. Centring this on 26.2 MHz the bandpass can be displayed directly on the oscilloscope. This technique was used extensively

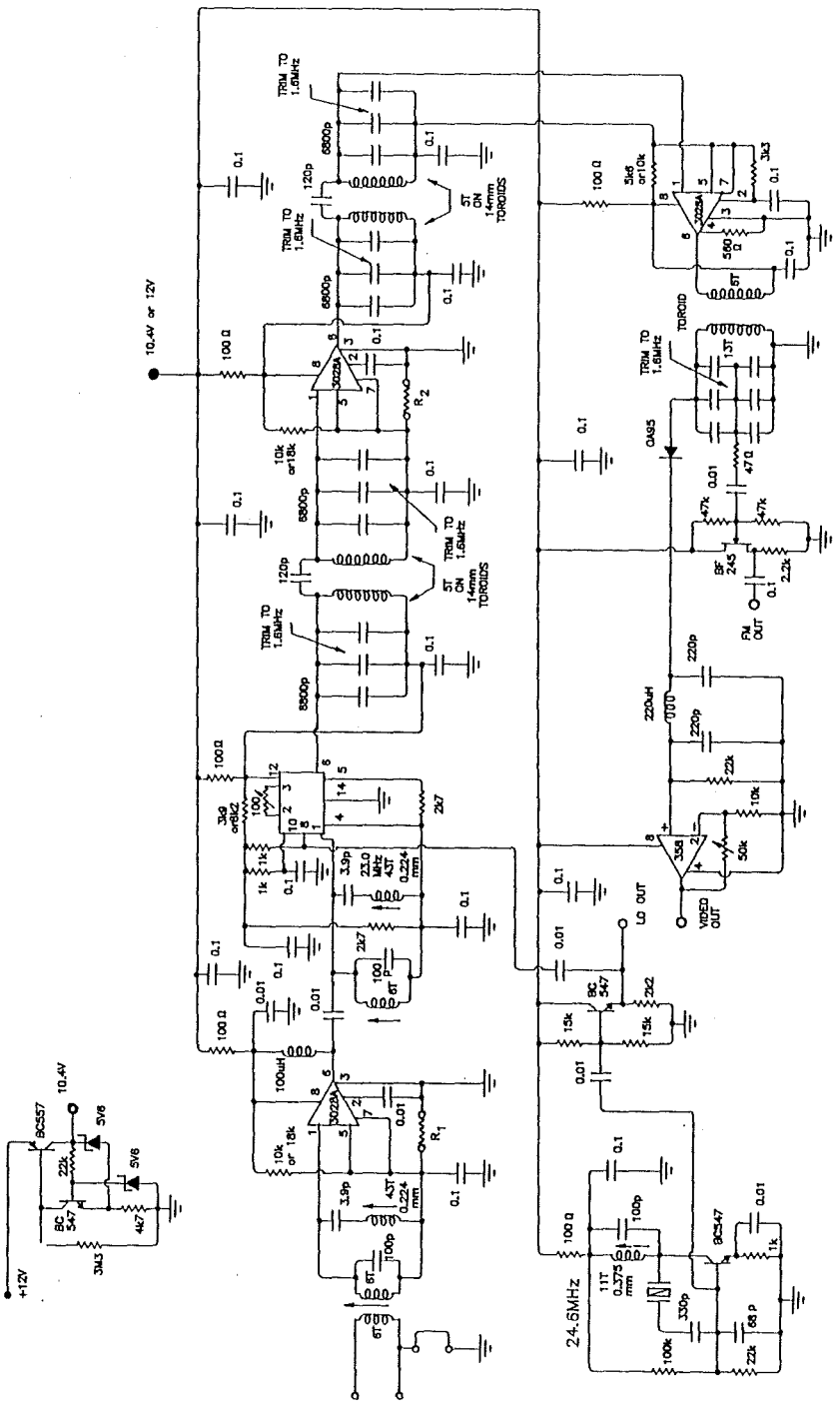


Figure 2.2: The circuit diagram for the 26.2 MHz receivers.

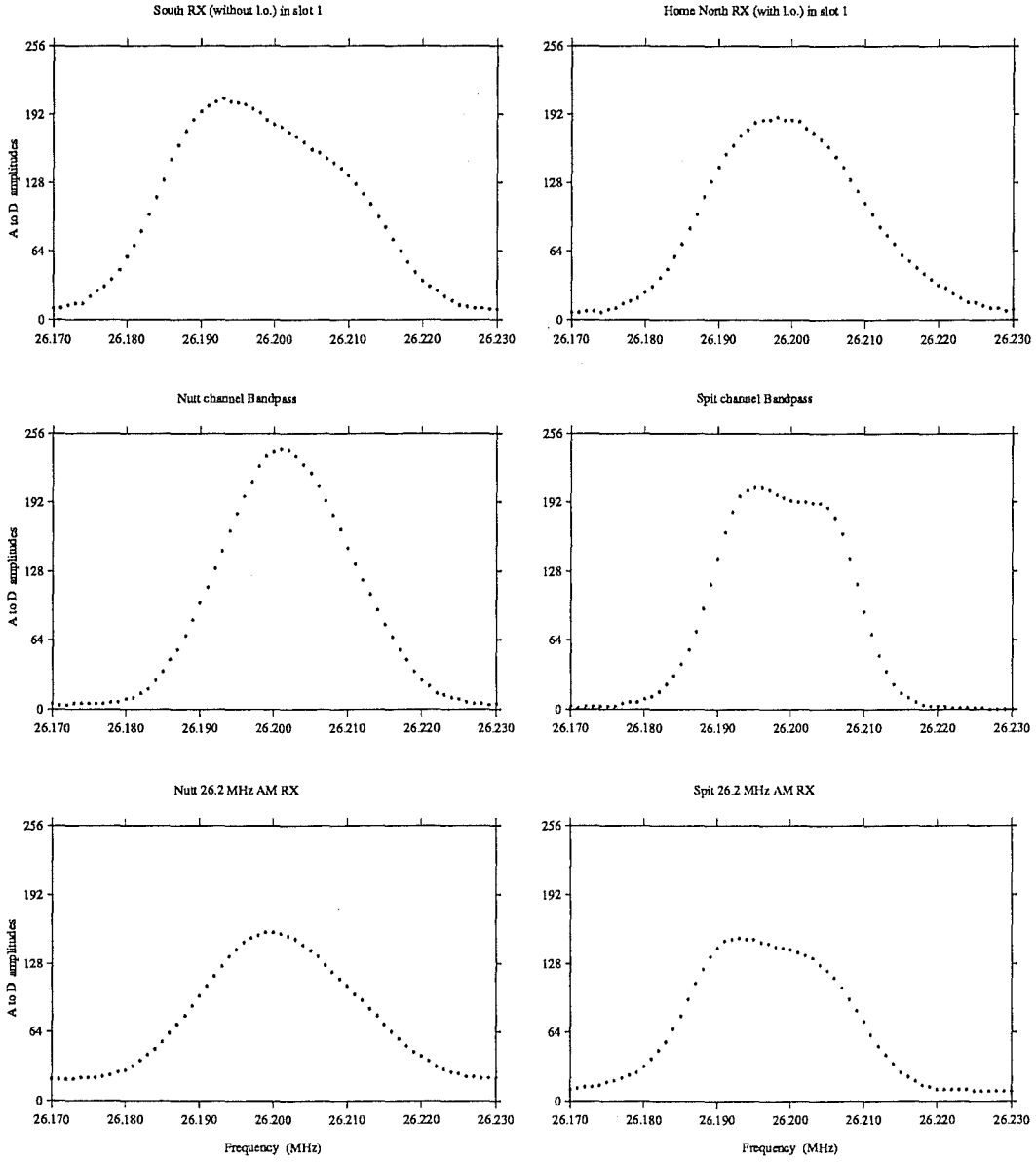


Figure 2.3: The bandpass response curves for the receivers. The curves were measured 1989 August 8. The frequency was shifted in 1 kHz steps, the video output measured through the A/D converters and recorded by the computer. The top four panels measure the receiver channels as they would appear at the Home site computer interface. The Nutt and Spit channel bandpass plots include transmission of the data through the FM links. Bandpass curves for the two remote site 26.2 MHz AM receivers are included for comparison.

in trying to improve both the shape and position of the receiver bandpass responses. The values of the trimming capacitors in the three stages on the IF strips were adjusted to improve the response. The result is a compromise between the narrowest possible bandpass, its symmetry and the proximity to 26.2 MHz. It was found the shape varied noticeably over the hour after the receiver was turned on complicating the adjustment[5]. A final trimming was left until the receiver had been running continuously for a hour.

Receiver Calibration

The receivers were calibrated using the Marconi signal generator mentioned above. The radio frequency signal strength was increased from 1 μV in 1 μV steps until the individual receiver channel was saturated. The results of this calibration are shown in Figure 2.4. The calibration curves for the video output of the 26.2 MHz receivers located at the remote sites are included for reference.

The FM data link introduces uncertainties in the amplitude of the remote site echo profiles. These are discussed in more detail in the chapter on data reduction. Figure 2.4 clearly shows the two gain settings of the remote site video outputs; once at the AM receivers and again at the FM link receiver. The Home site South receiver channel was chosen to give the reference amplitude for all meteor echoes detected. The calibration curve for this receiver is shown on an enlarged scale in Figure 2.5.

It should be mentioned here that the echo amplitudes recorded in the observation records are not direct measurements of the echo amplitude at that range bin. The rangebin sample could be at any position with respect to the peak echo amplitude. A quarter, half, quarter average on adjacent rangebins is recorded as the profile amplitude for that sweep. This recorded A/D amplitude is not the peak strength of the echo. It cannot be directly applied to the calibration curves in this section to give the radio frequency signal strength of the echo. Section 4.2.3 discusses the relation of recorded amplitude to peak echo amplitude.

Home Site Phase Comparison

The relative phase angle of the echo on each of the two Home site aerials needs to be measured. The two Home site receivers use a common local oscillator mounted in the North receiver. The two 1.6 MHz I.F. signals are used to make a phase comparison between them. This comparison is done by a third units mounted in the receiver rack. Each signal is fed into a phase locked oscillator which generates an essentially square pulse related to the phase of the signal.

[5]If memory serves me correctly there was some variation with signal amplitude to be included in this compromise as well. This work was undertaken in an effort to reduce the effects of radio interference on the system.

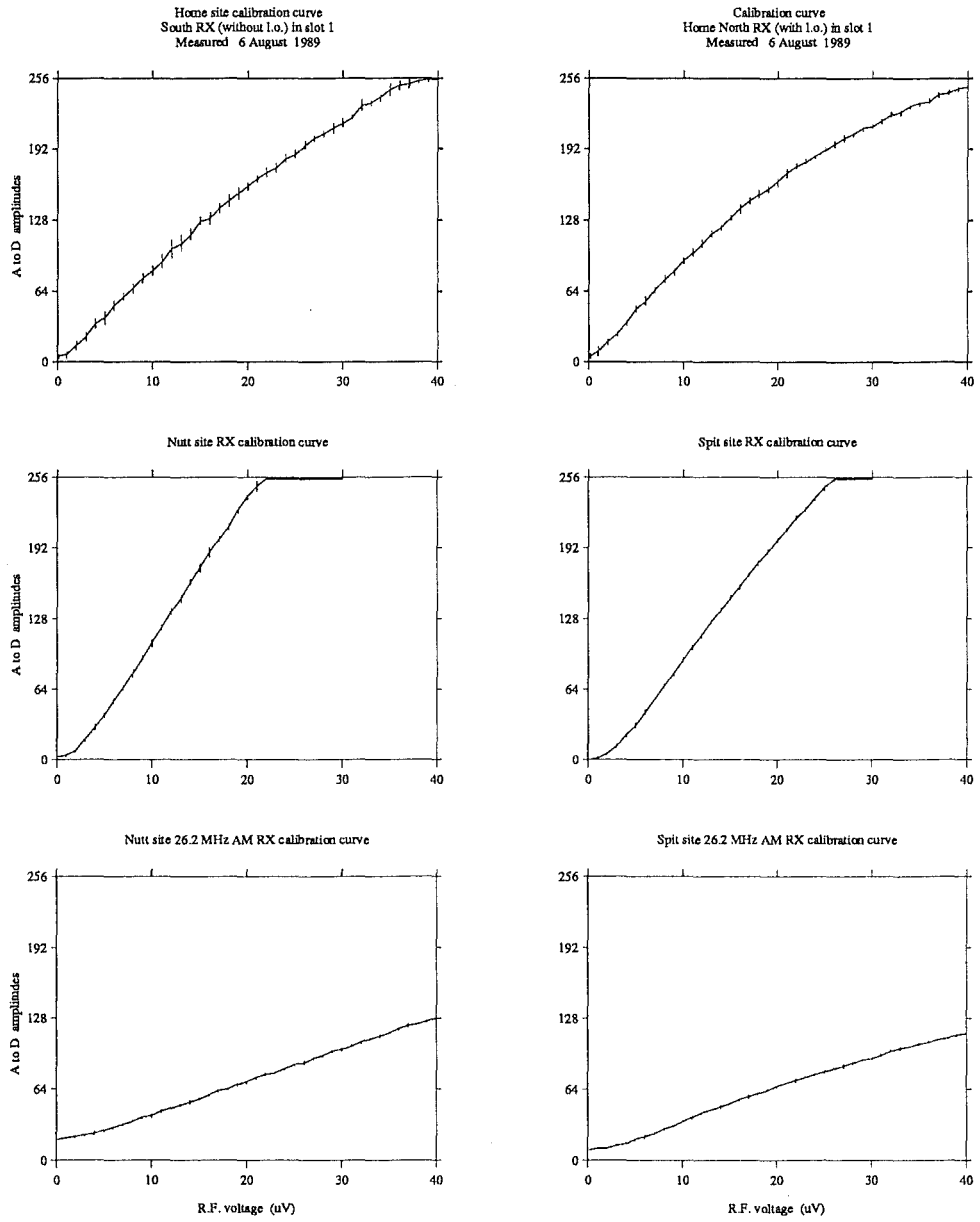


Figure 2.4: The calibration curves for the three receiver channels. The input signal strength is plotted against A/D amplitude of the video output voltage. The error bars plot the standard deviation of a series of samples taken on the stated signal strength. The calibration curves were measured on 1989 August 6.

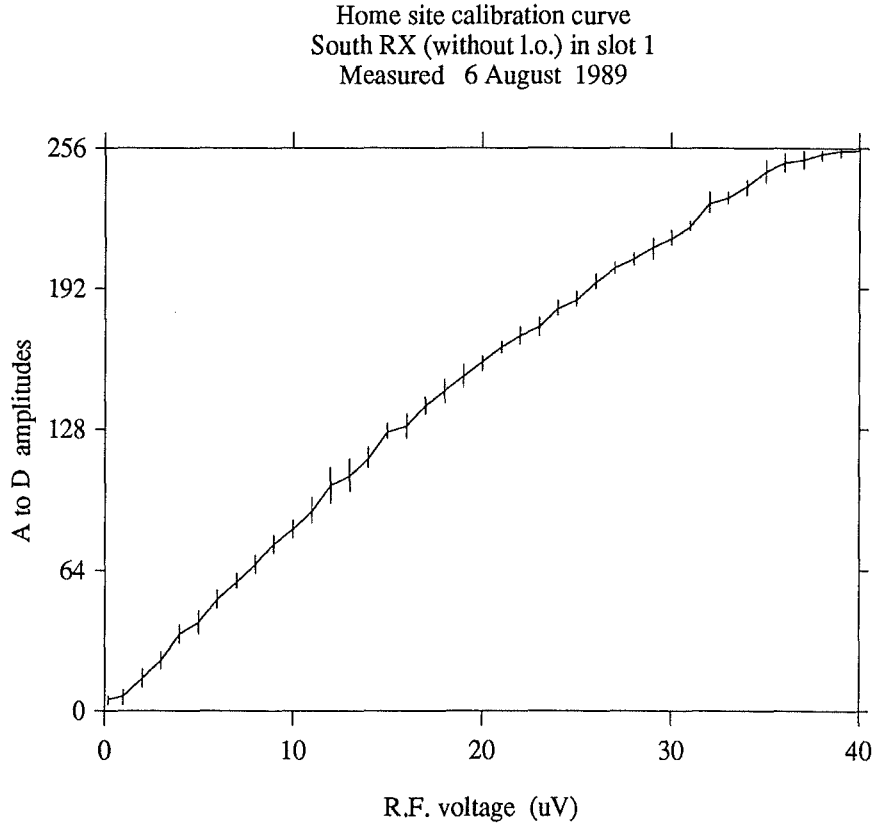


Figure 2.5: The calibration curves for the Home site south receiver channel. This receiver is used to provide the reference amplitude of the meteor echo. The figure is an enlarged copy of the same data in Figure 2.4.

Radio noise will lock the circuit but produce random results for the phase difference between the two input signals. A echo with reasonable amplitude will cause the loop to lock at the phase of the input signal.

The two signals are fed into phase sensitive detectors, the output of which will be related to the differential phase. Combining square pulses will produce a triangular output. One signal is shifted by 90° for the second detector to provide enough information to determine the quadrant of the phase. The calibration of these Tin and Tos voltage outputs are discussed in Section 4.3.1.

2.2 Aerials

The AMOR system includes a total of nine aerials. Four collinear receiver arrays and one rhombic transmitter array are designed to operate at the radar frequency of 26.2 MHz. The remaining four are the three element Yagi aerials of the two 39 MHz FM data links. The aerials have a broad enough band so that small changes in the operating frequency can be handled, for example from 26.36 to 26.2 MHz and from 39.6 to 39.0 MHz.

The original design of the system meshed the transmitter and receiver patterns together. The two aerials were arranged so that the first null of the transmitter pattern fell at the same point as the secondary maximum of the receiver arrays. This produced the narrowest effective beam with the shortest possible aerials. In retrospect the narrow beam should have been achieved using the transmitter array only. This would have reduced the labour in developing and maintaining four long receiver aerials. Broader arrays would ensure that all three receiver sites have the greatest likelihood of detecting an echo from any illuminated meteor trail.

The transmitter aerial produces a narrow beam in azimuth but with a broad distribution of power in elevation. Matching loads at the ends of the rhombics phased the array so most of the power was directed south. The position of the meteor trail could be assumed without too much danger of ambiguity[6]. The collinear receiver aerials give a narrow beam to both the north and south. In fact the timelags determined from echo amplitude profiles can unambiguously place the trails in either the north or south lobes. A new collinear transmitter array has been built and awaits commissioning. This will radiate equally both north and south. The detection of particles from polar orbits will be improved. This new array is considerably longer than the current transmitter aerial. A narrower beam with greater peak field intensity will be achieved. This should move the detection limit to a fainter magnitude. The accuracy in angular position of the meteor will be improved.

[6]The bulk of interplanetary dust is distributed near the ecliptic. Most meteors would therefore come from north of the Christchurch zenith. The appropriate reflection geometry for detection by the radar would place these meteor trails to the south of the station. With most of the transmitter power directed south this probability is increased.

1984 August 24λ coax array fed at six locations. Coax cable provided the active elements with the core and shields swapped every half wavelength.

1986 January Open wire 24λ collinear aerial with two feed points.

1986 November Remote site aerial relocated from the Swamp to Nutt site. 4λ single feed open wire collinear.

1988 November Nutt site extended to 12λ with one feed point.

1989 February Transmission line added and Nutt site brought in line with other 24λ aerials.

1989 August After measuring the power diagram associated with this configuration the receiver aerials were changed to an array of $6 \times 2\lambda$ bays. The two Home site aerials contained 8 bays.

1990 January The Spit site aerial was moved a further 4 km down the Kaitorete Spit.

Table 2.1: A short potted history of the collinear receiver arrays.

An estimate needs to be made of the measurement uncertainty in assuming the meteor detection points are on the meridian. Both the orientation and width of the beam determine this. Beam width contributes the biggest uncertainty. Crudely, the location is known to within the 3 dB half power points of the aerial array power distribution patterns. The radar magnitude of meteor echoes detected by the system depends on the gain of the aerial arrays. This gain is the ratio of the peak power radiated by the antenna to a theoretical equidirectional power distribution.

2.2.1 Collinear Receiver Arrays

The 26.2 MHz receiver aerials are composed of centrally fed 2λ collinear sections in a broadside array. The field station at Birdlings Flat is an extremely corrosive sea-side environment It is exposed to both southerly storm fronts and nor-west gales. Considerable work was done to make these receiver arrays reliable under these conditions. Table 2.1 gives a brief history of the receiver aerials.

A schematic diagram of the collinear arrays in their final form can be found in Figure 2.6. The collinear bays are all fed in phase at 2λ intervals. The signal travels down an open wire transmission line to a central feed point for the receiver. The two Home site receiver aerials are longer at 16λ with the addition of two extra bays. A detail showing the layout of one of the

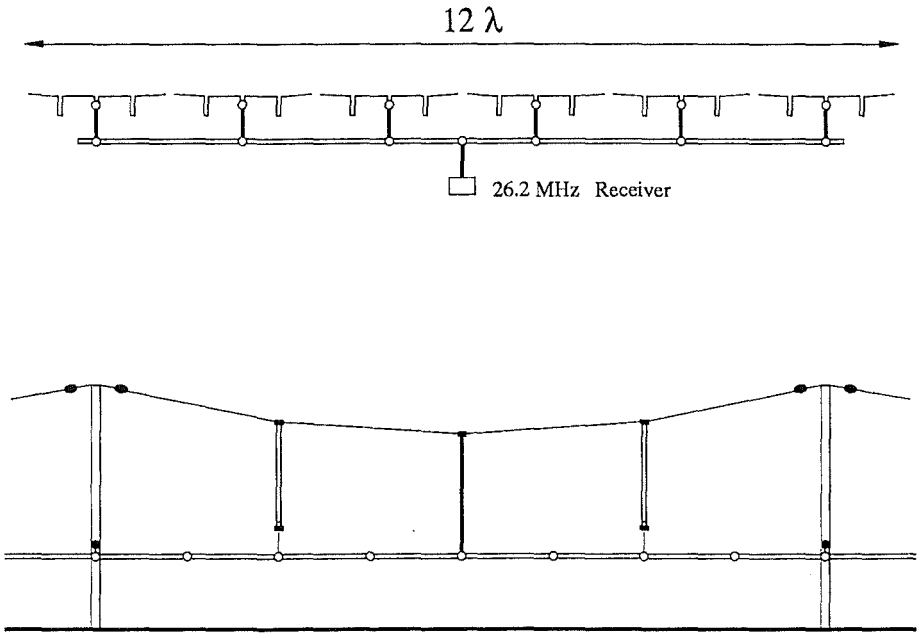


Figure 2.6: A schematic diagram of the collinear receiver array. The top sketch shows an elevation of the Nutt site aerial layout as it finally evolved. A detail of one collinear bay is drawn at the bottom. Insulator connectors are shown as filled symbols. Two twin coaxial cables join the aerial to the main transmission line. The matching stubs are secured to spacer rings in the transmission line with rope.

receiver bays is included in Figure 2.6. The solid rectangles represent perspex insulators. The bay is slung between 6.6 m high tubular steel poles. Egg insulators connect the aerial wire to the securing ropes. The perspex at the ends of the quarter wave matching stubs are secured with rope to ring spacers in the main transmission line. Separation of the wires in these matching stubs is ensured by plastic ring spacers. The transmission line is slung from ringbolts about 2.5 m off the ground. Two half wavelength long coax cables in a twin arrangement connect the transmission line to the radiating section. All wire to wire connections within the aerial are brazed. The open wire configuration of the aerial allows a visual inspection for electrical breaks. The 16 gauge hard drawn copper wire has reasonable mechanical strength and is self supporting. All connections with coax cable are soldered and then potted in epoxy before assembly in the field.

Theoretical Power Distributions

A meteor echo detected by a receiver can come from any direction. The power distribution of the aerial can be interpreted as a probability distribution for where the trail actually is. The narrow beam of the aerials is used to locate the meteor trails in azimuth on the meridian. Its width represents the uncertainty implicit in this assumption. A gain factor for the receiver aerial array can be calculated from the power distribution. This is necessary to estimate the radar magnitude of the meteor trail. This section discusses the theoretical power distribution for the aerial.

The collinear receiver aerials can each be treated as a filled broadside array of dipole radiators. This array is 16λ long in the case of the Home site aerials and 12λ for the remote site receiver arrays. The azimuth pattern will be dominated by the large number of multiple radiators. In elevation the collinear array is just a point source 0.6λ above the ground plane reflector. This gives the elevation power distribution.

The variation of the relative electric field in the azimuthal plane is expressed in equation (2.1). This is done for the 16λ Home site array. The electric field strength H, is expressed as a function of the azimuthal angle θ. It is the product of the half wave dipole distribution and an array factor.

$$H(\theta) = \frac{\cos(\frac{\pi}{2}\sin\theta)}{\cos\theta} \times \frac{\sin(16\pi\sin\theta)}{\sin(\frac{\pi}{2}\sin\theta)} \tag{2.1}$$

The relative gain in any direction is proportional to the square of this electric field strength.[7] A diagram showing the distribution that results from this relation is plotted in Figure 2.7(a). The first nulls occur at $\pm 3.58°$. 91% of

[7]The gain of the antenna is discussed as if it were being used for transmission since this makes the argument rather more direct, but by reciprocity the results apply to reception also. The power distribution of the Nutt site array was in fact measured by attaching a small transmitter to the antenna.

the received power is comes from the main lobe at right angles to the array. Calculation showed that the half power 3 dB points occur at $\pm 1.59°$. The distribution is symmetric about $\theta = 0°$. The relative gain distribution to the north ($\theta = 180°$) is identical.

In a plane perpendicular to the axes of the dipoles the radiation distribution is uniform and for a long collinear array all significant radiation is near this plane. The electric field as a function of elevation ψ, for such an array will depend entirely upon the reflecting ground plane. The radiating sections of the collinear array are slung approximately 0.6λ above the ground which is assumed to be a perfect reflector.

$$H(\theta) \quad = \quad 2\sin(0.6 \times 2\pi \sin\psi) \tag{2.2}$$

The variation in power distribution with elevation is plotted as Figure 2.7(b). This shows a broad beam centred at about 24.6° elevation. The 3 dB half power points occur at 12.0° and 38.7°. In this cross section a relatively large 19% of the power is directed into the lobe centred on the local zenith. In fact this is not as severe as it appears since the solid angle associated with power toward the zenith is reduced. Almost certainly the ground plane reflecting surface is not located at the 'grass' level[8]. Increasing the height above the ground in equation (2.2) narrows the beam in elevation and increases the relative strength of the zenith lobe. The aerials are suspended from 6.6 m poles and so the 0.6λ estimate is conservative anyway. Reducing the height of the radiating array to half a wavelength would remove the zenith lobe entirely. A broader pattern in elevation with no nulls would be ideal.

The directivity of the aerial is the ratio of the peak power emitted in the direction desired compared with what it would be if the power were emitted uniformly in all directions. This directivity is the same as the antenna gain provided all the power is radiated by the the aerial. In a simple sense the radar beam can be considered as the solid angle between the 3 dB half power points. By assuming all the power is radiated in this beam an estimate of the gain can be made. Since the azimuthal power distribution is very narrow the solid angle can be estimated by multiplying the angular distance between the half power points for each of the azimuth and elevation power distributions. A complete sphere of 4π steradians is equivalent to 41253 square degrees. A simple estimate for the gain of the receiver aerial array is made in equation (2.3). Equal power is radiated to both the north and south requiring an extra factor of 2.

$$G_R \quad = \quad \frac{41253}{2 \times (26.7 \times 3.18)} \quad = \quad 243 \tag{2.3}$$

A more precise estimate of the gain can be achieved using the theoretical

[8]In fact very little grass grows at the Birdlings Flat field station. The substratum is almost entirely shingle and larger stones. Very little water is supported near the surface.

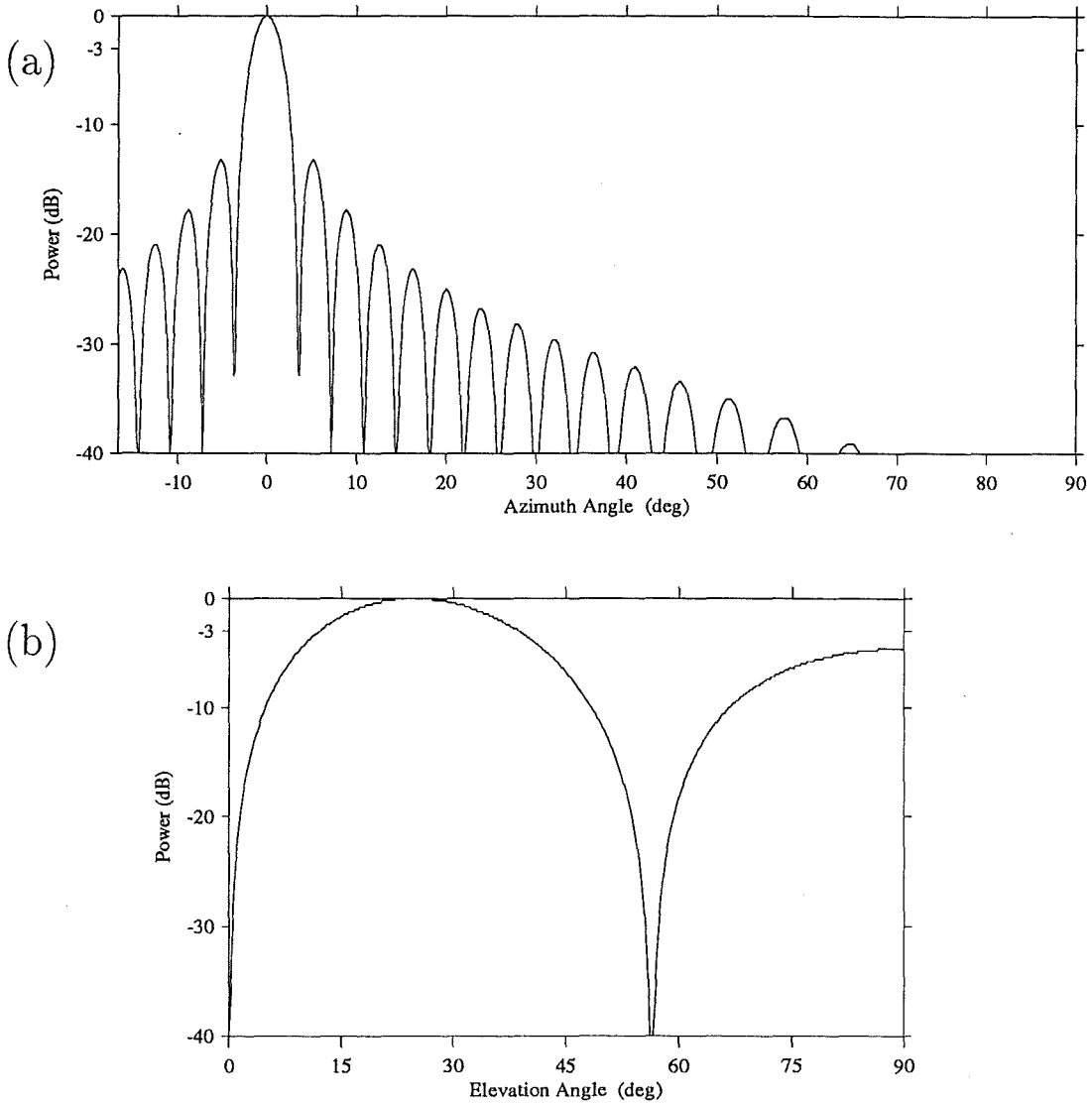


Figure 2.7: The theoretical power distributions for the 16λ long Home site receiver arrays. The variations were calculated from the electric field strength equations (2.1) and (2.2). (a) The azimuthal pattern is symmetric about 0°. The pattern is identical to both the north and south. Peak power occurs at 0° with 3 dB half power points at $\pm 1.59°$. The diagram is calculated for an elevation of 23° although since it is normalised this makes no difference. (b) Elevation power distribution; maximum 24.6°, 3 dB points 12.0° and 38.7°.

power distributions.[9] The power radiated in all azimuthal directions can be integrated from equation (2.1). The *peak* power if this were uniformly transmitted in azimuth can then be estimated. The power gain of the aerial in azimuth is the ratio of this value to the actual peak power. A similar procedure is adopted to calculate the power gain in elevation. Again since most of the power is concentrated in a narrow beam multiplying these two components to give an overall aerial gain is justified. The integrations were done numerically in 0.1/degrees steps. In the case of the azimuthal pattern 91% of the power was concentrated in the main lobe between $\pm 3.58°$.

$$\begin{aligned} G_{\mathrm{R}} &= G_{\mathrm{Azimuth}} \times G_{\mathrm{Elevation}} \\ &= 49.3 \times 2.7 \;=\; 133 \end{aligned} \tag{2.4}$$

The remote site aerials are only three quarters the length of the Home site receiver aerials. The azimuth power distribution is correspondingly broader and the gain for these by the same method is $G_{\mathrm{Remote}} = 104$.

Measured Power Patterns

The expansion of the Nutt site aerial to a full 24λ array in February 1989 offered an opportunity to physically measure the power distribution of the collinear arrays[10]. A small 26.2 MHz transmitter was attached to the Nutt site aerial. A dipole aerial and receiver were used to measure the signal strength along Bayleys Road. This road runs east-west and is situated on the opposite side of the Lake Ellesmere almost exactly 7 km south of the remote site. Figure 2.8 shows the results of the power measurements made in July 1989. The small stockwater windmill on Bayleys Road was used as a benchmark to locate the field strength measurements.

Figure 2.8(a) was the first power distribution diagram measured. Geographic south corresponds to the dip in signal strength at around +0.68 km. In hindsight the power distribution is suggestive of a double slit interference pattern. At this stage the Nutt site aerial was fed from two points 12λ apart. Having each of the two sections 180° out of phase would produce a double peak with a central minimum. Visual checks suggested this was not the case but in an experimental spirit, one wing was reconnected with its phase changed by 180°. Figure 2.8(b) was the result. The same sort of interference pattern occured but with maxima and minima swapped. The periodicity in the two patterns is about 550 to 600 m which at the 7 km distance of Bayleys Road

[9]The difference between the two estimates of antenna gain are rather large. there does not seem to be any obvious reason for this.

[10]The combination of sea, lake and mountains make it almost impossible to measure patterns for aerials at any of the other sites. Moving outside the near field and staying dry being mutually exclusive. The preliminary measurements were done in conjunction with Roy Seaton as part of a Stage III research project. Imagine our surprise when the results were exactly, not what was expected.

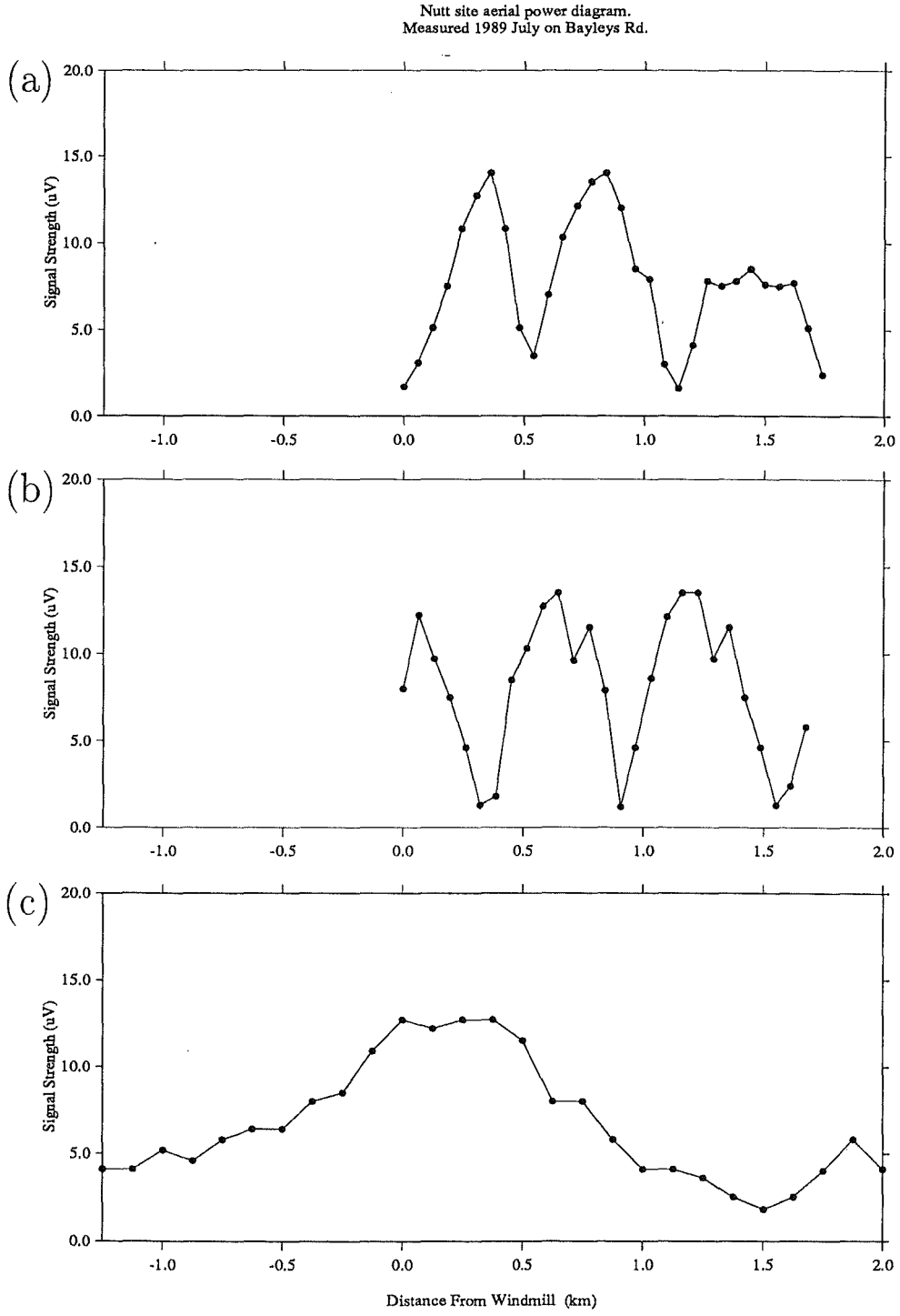


Figure 2.8: The original collinear receiver arrays had only two feed points. These power diagrams show the aerial patterns as measured on Bayleys Road, 7 km south of the Nutt site. Due south from the aerial is about +680 m West from the windmill. The receiver has a DC offset of 1 μV.
(a) The original 24λ array with two feed points, July 8.
(b) Original array with 180° phase change on one feed, July 14.
(c) Half the aerial disconnected, one feed point, July 18.

from the array corresponds to an angular separation of 4.7°. For a one wavelength path difference a slit separation of 12.2λ is indicated. The two feed points seem to be acting as separated transmission sources.

Figure 2.8(c) shows the power diagram from just one 12λ wing of the aerial. The pattern shows a central maximum. There are however no clear nulls and subsidiary maximums that would be associated with an ideal single slit pattern. The distribution implies a falloff in power radiated from aerial sections further from the feed point. Supposing that all power was radiated at the same intensity along the aerial an estimate of how much is *active* can be made. The half power points (0.707 × peak amplitude) indicate about 8λ is active. Assuming a first null at +1.5 km and a maximum at +1.9 km the active section is 5.6λ and 6.5λ respectively. Clearly the current distribution in the aerial drops away too rapidly for two fed points to be adequate. We suspect the perspex insulators used to support the matching sections are a major factor in the current falloff.[11]

The aerial should be fed more frequently than the *active* section implied by the analysis above. It was decided each 2λ bay should be fed individually as described earlier. The Nutt site aerial was reconstructed along with the three other receiver aerials. The power diagram was measured and the results are presented in Figure 2.9. Here the same analysis as above indicated that about 12.5λ of the aerial is *active*. The 3 dB half power points give an angular separation of 4.4°. The theoretical azimuth power distribution for the 12λ remote site aerial arrays predicts a separation of 4.3°.[12]

Measuring the power diagram also provides an opportunity to check the orientation of the aerial beam. From the measured pattern the beam is directed due south. Realistically the peak can be placed to an accuracy of about 50 m. This corresponds to a measurement uncertainty of about 0.4°. The east-west alignment of the aerials was checked using a theodolite with the Southern Alps (150 km distant) as a reference marker. This geometric alignment was also checked by the meridian transit of a number of stars. All the aerial arrays were geographically aligned in the same manner. The fact that 60% of the total meteor observations have echoes detected on all three stations indicates a substantial overlap of the beam patterns. It is reasonable to conclude that all the receiver beams are aligned north-south to better than half a degree[13].

[11] With more complete angular coverage for the power distribution a Fourier transform would yield more detailed information on the current distribution. As noted earlier, geography renders this impossible.

[12] Happiness is a well behaved aerial.

[13] The hope is that the transmitter aerial beam is also aligned to better than 0.5°. Measuring the transmitter power distribution using an observer positioned on the STS *Spirit of New Zealand* has not been realised to date.

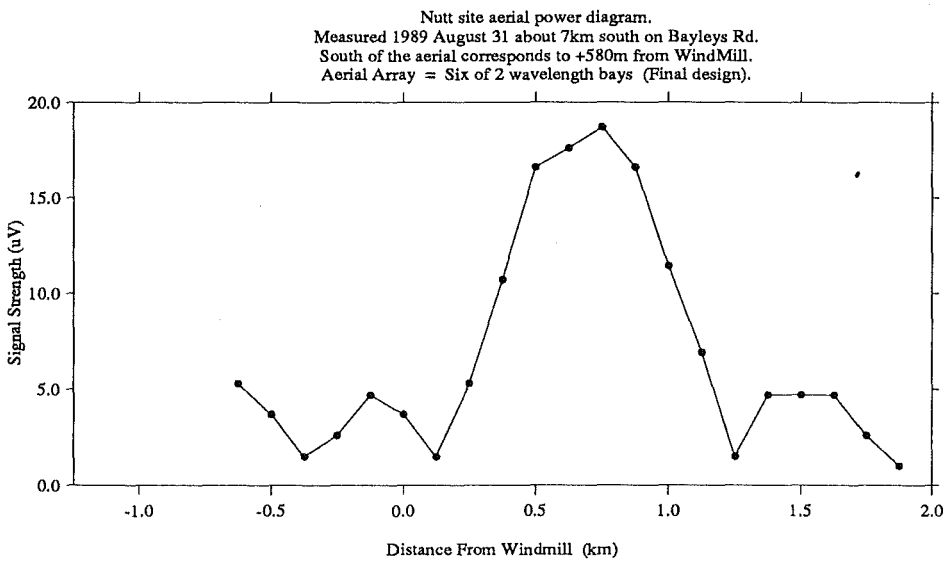


Figure 2.9: The power distribution for the final configuration of the remote site receiver aerials. This Nutt site distribution was measured 7 km from the aerial along Bayleys Road. Due south from the aerial is about +680 m West from the windmill. The receiver has a DC offset of 1 μV.

Celestial Sources

Ideally the aerial power distributions should be measured above the ground plane. One possibility is to look for the transit of celestial radio sources through the aerial beams. These would need to be sufficiently strong to be detected above the general radio sky noise. To improve detection of these weak radio sources a special observing program was developed to integrate noise level measurements. The known transit times of the celestial radio sources would provide a more conclusive check on the correct alignment of the respective beams.

An estimate of the expected noise level is necessary to identify candidate sources. The receiver noise at 26.2 MHz is a about 10 dB less than sky noise. Only the latter is considered. The noise power is given by equation (2.5) where $k = 1.38 \times 10^{-23}$ J.K^{-1}, is Boltzmann's constant and Δf the receiver bandwidth.

$$P_{\text{Noise}} = kT_N\Delta f \tag{2.5}$$

The equivalent sky noise temperature, $T_N \simeq 3 \times 10^4$ K, is given by McKinley (1961) for 26 MHz. The receivers have a 20 kHz bandwidth with a 50 Ω input impedance. The expected noise strength therefore follows.

$$V = (2RP_N)^{\frac{1}{2}} \simeq 1\mu V \tag{2.6}$$

This is approximately what we measure with our receivers.

Hey (1949) gives decameter wavelength radio intensities for the quiet Sun at around 10^{-22} W.m^{-2}.Hz^{-1}. Solar storms which last from several hours to a few days are a factor of 10^2 more intense. The equivalent area for an antenna can be calculated from equation (2.7) (McKinley, 1961).

$$A = \frac{\lambda^2 \times \text{Gain}}{4\pi} \tag{2.7}$$

For the radar wavelength of 11.45 m and the Home site receiver aerial gain of 133 the equivalent area is $A \simeq 1400$ m^2. The power can then be estimated as $P_N = IA\Delta f$ for the two different solar conditions. Signal strength follows from equation (2.6). For a quiet Sun a $2\mu V$ signal is indicated and therefore under normal conditions the Sun's motion through the beam will barely be detectable. Indeed no appreciable increase in the radio noise was detected. The signal strength would rise to about $17\mu V$ during storm conditions. Using the sun as a celestial source would be possible during solar storms.

Jupiter is an intense source of decameter radiation. Its output is modulated by the relative position of Io and is very complex. A typical range of intensities is $I \simeq 5$ to 20×10^{-21} W.m^{-2}.Hz^{-1} (at 27.6 MHz) from Allen (1973). Even at the lower value this gives a signal strength of $V \simeq 4\mu V$ which should be discernible. Below is a list of other possible discrete celestial sources (Allen,

1973). All of them are of lower intensity than Jupiter but their emission intensities are more constant. All these sources should give signal strengths of one to two microvolts and might be detectable.

	Intensities ($W.m^{-2}.Hz^{-1}$)
Cas A	5×10^{-22}
Tau A	6×10^{-22}
Cyg A	3×10^{-22}
Vir A	6×10^{-23}

Despite several attempts to identify celestial sources moving through the receiver beams no convincing enhancement in the received signal strength was detected. Slow drifts in the noise levels, especially of the remote sites were detected. Periods of increased noise seemed to be uncorrelated with the expected procession of celestial sources across the sky. The integration program could usefully be improved and this test rerun.

2.2.2 Transmitter Array

The transmitter antenna is a broadside array of eight rhombics fed in phase. Figure 2.10 provides a schematic layout for the array. Each leg of the rhombic is 2λ long and is supported at its ends 6.5 m above the ground. Neighbouring rhombics share a common pole. Each rhombic is terminated by a 680Ω load composed of 20 carbon resistors in parallel and mounted in a weather case. The loads dissipate about 25% of the power put out by the transmitter. They ensure the aerial is phased to direct the main lobe toward the south. Each rhombic is fed in phase at 2λ intervals. A half wavelength long open wire transformer connects the radiating section to the feeding transmission line which is supported 2.3 m above the ground on an independent set of poles.

The gain of the transmitter aerial array needs to be estimated. Bruce *et al* (1935) give the theoretical radiation pattern for the electric field pattern, H, of a rhombic aerial. Since the receiver aerials are horizontally polarised, only the horizontal component of the distribution need be considered:

$$H(\theta,\psi) = \frac{\sin(\pi K_1 l/\lambda)}{(\pi K_1 l/\lambda)} \times \frac{\sin(\pi K_2 l/\lambda)}{(\pi K_2 l/\lambda)} \times [\cos\theta - \sin\phi\cos\psi] \quad (2.8)$$

$$K_1 = 1 - \cos\psi\sin(\phi+\theta) \quad (2.9)$$

$$K_2 = 1 - \cos\psi\sin(\phi-\theta) \quad (2.10)$$

Where the azimuth and elevation angles are given by θ and ψ respectively for the transmitter array with rhombic leg length, $l = 2\lambda$, and apex angle, $\phi = 60°$ can be substituted. The electric field pattern in azimuth is the product of the single rhombic pattern and an array factor, equation (2.11). The distribution in elevation is a combination of the vertical rhombic field and a ground factor,

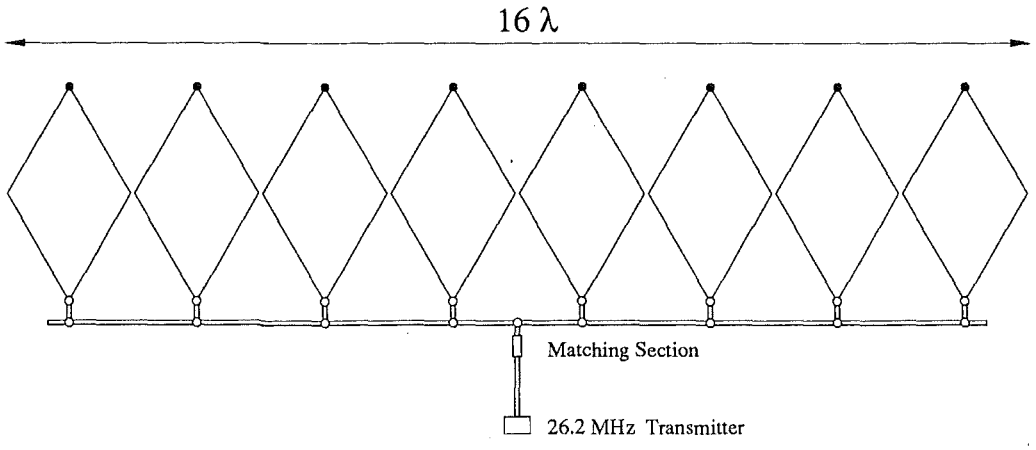


Figure 2.10: A schematic diagram of the rhombic transmitter array. The rhombics are fed at 2λ intervals from the transmission line. The filled circles represent the 680Ω loads used to phase the individual rhombics. The aerial is shown in plan view. The main beam is directed south toward the top of the page.

equation (2.12). The aerial is suspended from 6.6 m poles. A height of 0.6λ is assumed.

$$H(\text{Array Factor}) = \frac{\sin(16\pi \sin\theta)}{\sin(\frac{\pi}{2}\sin\theta)} \quad (2.11)$$

$$H(\text{Ground Factor}) = 2\sin(0.6 \times 2\pi \sin\psi) \quad (2.12)$$

The power distributions in azimuth and elevation are graphed in Figure 2.11. In this idealised case the peak power of the north lobe is 20 dB less than the main south directed beam. This assumes perfect power dissipation by the rhombic loads with no RF reflection.

Assuming all the power goes into the main beam between the 3 dB points the gain for the array can be estimated.

$$G_T = \frac{41253}{20.7 \times 3.14} = 635 \quad (2.13)$$

This figure provides a crude estimate and almost certainly overstates the aerial gain. As with the receiver aerials most of the power is concentrated in a narrow azimuthal beam. It is therefore reasonable to calculate the gain in each of the azimuth and elevation cross sections and multiply the results to give the overall aerial gain. This has the advantage of greatly simplifying the numerical task of integrating the power distribution. This integration was carried out yielding

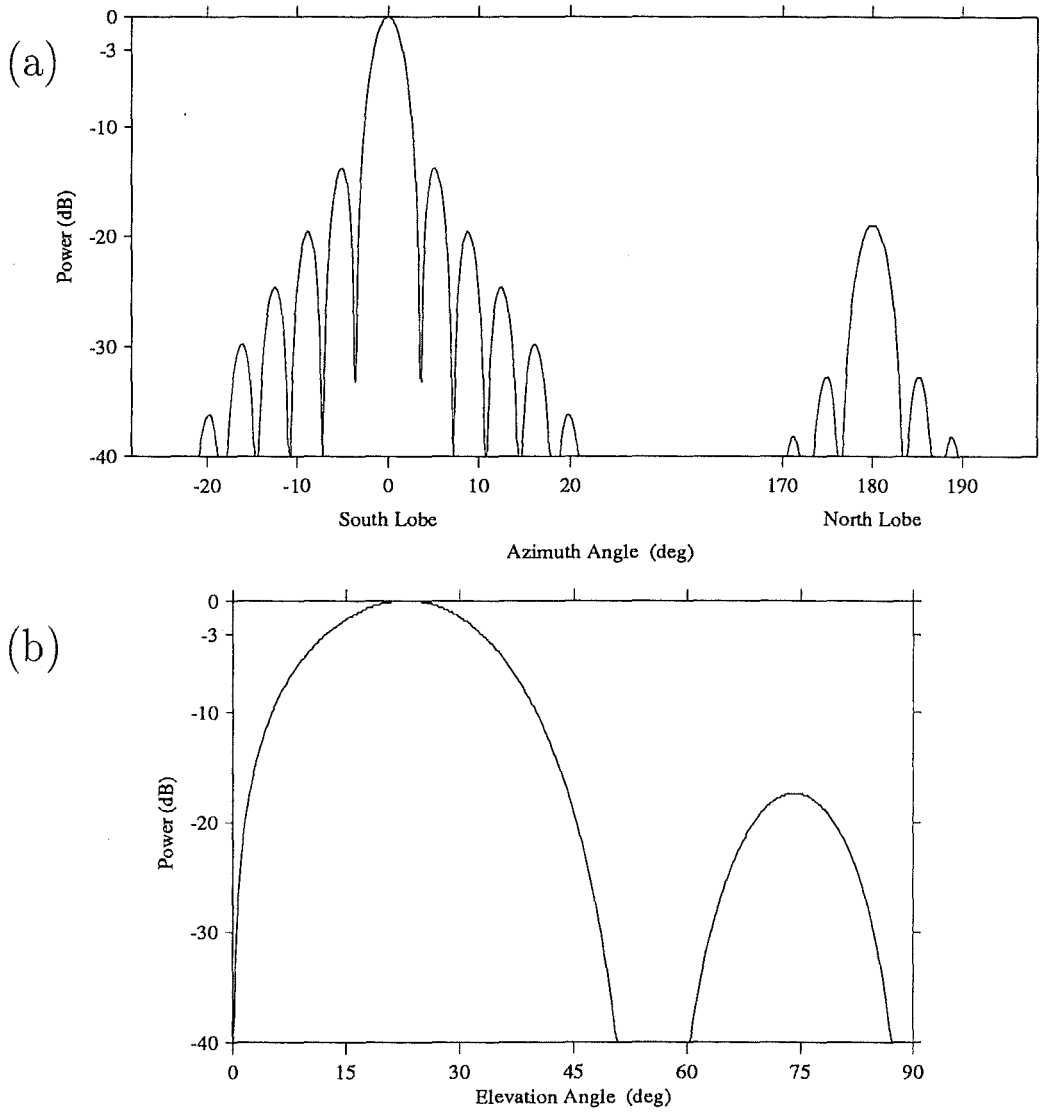


Figure 2.11: The theoretical power distributions for the transmitter aerial array. (a) The azimuthal pattern gives the main peak due south. The half power 3 dB points occur at $\pm 1.57°$. 94% of the power is included in the main south lobe. The use of rhombics produces a more rapid drop off in power for the secondary lobes compared with the collinear receiver arrays. (b) The broad power distribution in elevation peaks at 22.9°. Half power points occur at 12.3° and 33.0°.

a gain factor for the transmitter array.

$$\begin{aligned} G_{\mathrm{T}} &= G_{\mathrm{Azimuth}} \times G_{\mathrm{Elevation}} \\ &= 100.5 \times 4.25 \;=\; 427 \end{aligned} \tag{2.14}$$

2.3 AMOR Magnitude Limit

The radar equation (2.15) for the echo power P_{Echo}, in terms of the system parameters is derived by McKinley (1961). The variables are defined in tabular form.

$$P_{\mathrm{Echo}} = \frac{P_T G_T G_R}{64\pi^3} \left[\frac{\lambda}{R_o}\right]^3 \sigma_e q^2 \tag{2.15}$$

P_T	Transmitter pulse peak power, $\frac{3}{4} \times 20$ kW.
G_T	Transmitter aerial array gain, 427.
G_R	Receiver aerial array gain, 133 (Home site).
λ	Radar wavelength, 11.45 m.
R_o	Typical range to the meteor trail, 200 km.
σ_e	Scattering cross section of an electron, 10^{-28} m^2.
q	Line density of electrons, underdense trail.

The terminating loads on the rhombic array should theoretically dissipate 25% of the transmitter power. This dissipation would ensure no reflection from the apex of the rhombic. Any reflections would cause power radiated into the back lobe. In consequence the transmitter pulse peak power needs to be reduced by one quarter. The lower amplitude limit for meteor echoes from which orbits can be determined is about $4\mu V$ or 30 A/D units. This gives a voltage signal to noise ratio of about 4. For the 50Ω impedance of the receivers this corresponds to an echo power of 1.6×10^{-13} W. The line density of the ionisation column for this echo strength is $q = 1.4 \times 10^{11}\mathrm{m}^{-1}$. McKinley (1961) for non-shower meteors relates the line density to a radar magnitude.

$$M_r = 42.4 - 2.65 \log_{10} q \tag{2.16}$$

This gives an equivalent radar magnitude of +12.9 mag. [14]

As a comparison, halving the operating power of the transmitter and using the receiver aerial gain for the remote sites, the line density for a successful orbit determination rises to $2.5 \times 10^{11}\mathrm{m}^{-1}$ or a radar magnitude of +12.2. Increasing the transmitter output to nearer full power would give an average power of 700 W or 28 kW peak power. An electron line density of $1.2 \times$

[14] $q = 2 \times 10^{14}\mathrm{m}^{-1}$ gives $\mathrm{M}_r = +5$ mag. $q = 2 \times 10^{12}\mathrm{m}^{-1}$ gives $\mathrm{M}_r = +10$ mag.

$10^{11}\mathrm{m}^{-1}$ or radar magnitude of +13.1 is indicated. In general the limiting radar magnitude of the AMOR system should be taken as +12.5 magnitudes.

It is an interesting exercise to convert these limiting magnitudes to physical sizes. Let us assume a zero magnitude meteor to have a mass of 1g. A plausible relation of radar magnitude to mass is given here.

$$M_r = -2.5 \log_{10} m \tag{2.17}$$

This gives a +5 mag meteor 10^{-2}g mass, and +10 mag as 10^{-4}g. Also assuming a particle density of 10^3 kg.m^{-3} and an approximately spherical shape the limiting magnitude of the AMOR system corresponds to a particle size of 130 μm. Using the radar at close to maximum power the limiting size drops to 100 μm.

2.4 FM Data Links

Two frequency modulated data links send the echo amplitude data from the remote receiver sites to the data logging system at the main Home site. These links are required to transmit the data over 10.5 and 8.2 km from the Nutt and Spit sites respectively. The links operate on 39.0 MHz and horizontal polarisation for the swamp link from the Nutt site. The Spit link uses a vertical polarisation and 39.3 MHz. The remote site stations are powered by 40 Amp.Hour sealed gel batteries. Most of the power is used to drive the link transmitters.

Transmission strength along the ground wave was found to be better using vertical polarisation. The two different polarisations were chosen to avoid cross talk between the links. We have not experienced any problems with the 300 kHz separation. The vertical polarisation was originally allocated to the longer link. With the redeployment of the remote stations at new sites it was easier to retain the original polarisations.

The link aerials are three element Yagis constructed of aluminium tubing. The elements are all a little over 7.5 m long. These were originally supported in the middle to a horizontal steel strut. Wind induced vibrations caused an annoying number of fatigue cracks with the subsequent loss of half the element. The situation was rectified by welding 400 mm long half round sections directly to the strut and then securing the elements lengthwise along these.

The aerials are mounted to aim at the opposite ends of the link to within a few degrees. The Yagi antennas have a relatively broad horizontal distribution of radiated power so this is not critical. The aerials give some enhancement to the signal strength of the link system. Unfortunately the 11 kV mains supply transformer for the receiver hut is close to the receiver building and link receiver antennas. The 39 MHz link receivers seem most susceptible to corona bursts. The original link antennas were three element Yagis but multi-element Yagi have been substituted to narrow the receiver power distribution away from the transformer. A reduction in corona interference was achieved.

The FM transmitters consist of a voltage control oscillator (VCO), mixer and power amplifier. The video signal from the 26.2 MHz receiver is used to drive the VCO. This was centred on 1.6 MHz and produces a swing of ±40 kHz. Lower input voltages produce a higher frequency. The VCO output is combined with a local oscillator signal at 37.4 (or 37.7) MHz in the mixer. The signal is amplified in a PA stage. The transmitters have a power output of around 0.7 Watts. This is matched to a 50Ω impedance coaxial transmission line.

The transmitter unit was designed around the VCO on an XR215 chip available at the time. These have relatively poor stability with temperature variations. The chip is mounted in a thermally insulated oven that controls the temperature to within about 0.1°. The oven is maintained at 35°C; this temperature being a little above the daytime maximum expected for Christchurch. During development of the Nutt link short interference bursts at 11s intervals were detected. The source of these bursts was finally tracked to the thermal control circuit. It was discovered when the Nutt site transmitter was placed in a temperature controlled test box. The interference was reproduced as the heater circuit switched the oven element off and on to follow the environment changes. A noisy resistor was replaced and the problem rectified.[15]

The VCO was originally centred to give 1.6 MHz for 2.5 volts video input. This circuit would swing both positive and negative from this input. In fact the response of the VCO circuit was strongly non-linear through its zero swing. This was discovered while checking the calibration curves of the remote site channels through the FM links. Figure 2.13 shows the step in the calibration curves caused by this problem. It most obviously produced steps in low amplitude echoes on the Nutt site channel. Figure 2.12 gives an example that was discovered while testing the despiking routine (see section 4.2.1). To eliminate the problem the VCO was recentred to only swing one way. This forced a halving in the 26.2 MHz AM receiver video output range. The FM link receivers were adjusted to maintain a 0 to 10 volt dynamic range but with a reduced amplitude accuracy.

The link receivers use a 100 kHz bandwidth. Of this, 80 kHz was designed to be available for the frequency modulation. The need to swing the VCO one way only cut this to 40 kHz. The output frequencies of the link are noted below both in terms of input voltages to the link transmitter alone and RF signal strengths to the 26.2 MHz AM receiver with it connected. With the

[15] In searching for a source to this interference it was discovered a electric fence connection a kilometre from the Home site gave bursts of 26.2 MHz noise at 11s intervals. In hot pursuit of this the fence controller was found to reside in a pump shed. The pumping gear seemed to be switched on a periodic basis, at, as was subsequently discovered, a period slightly off the 11 s cycle. It would seem that the two sections of the electric fence on either side of a shelter belt acted as a large aerial picking up the interference on the 39 MHz signal being sent from the remote site. The intermittent 11s *flapping* of the Nutt site link signal remained a problem for two years before being finally isolated.

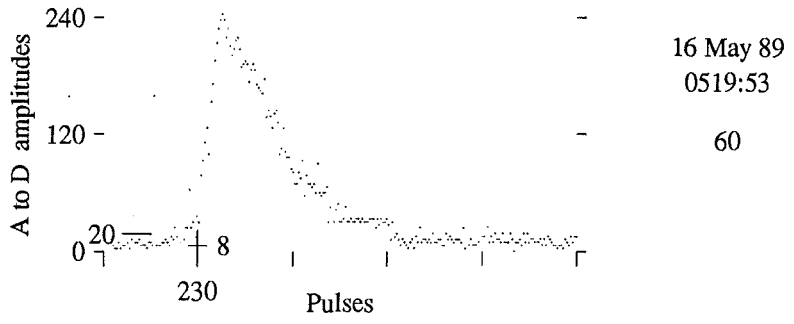


Figure 2.12: The effect of the VCO discontinuity on amplitude profiles is clearly demonstrated by this profile. It was discovered while testing the despiking routine. 8-bit echo amplitudes are plotted against the 250 radar pulses of an echo profile.

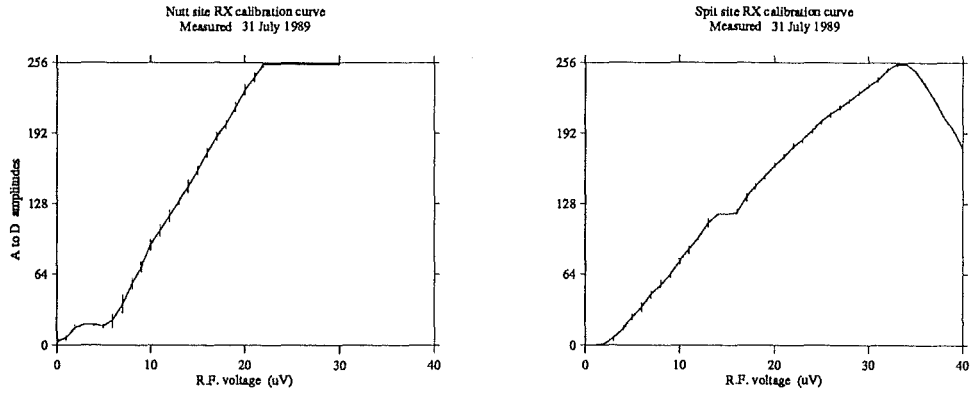


Figure 2.13: Calibration curves for the two remote site channels prior to 1989 August. The non-linear response of the FM transmitter VCO chip can be clearly seen. It seems to be associated with the section corresponding to zero swing. This is set in the middle of the Spit site amplitude range. For some reason this is not the case with the Nutt link. The Spit curve shows the link frequency being driven outside the FM receiver bandpass for input RF signal strengths above 34 μV.

AM receiver connected this latter case implicitly includes any DC offsets in the receiver output. Frequencies are given in MHz.

Nutt site FM TX frequencies

0 volts	39.393	0.2 μV	39.385
3 volts	39.351	20 μV	39.350

Site site FM TX frequencies

0 volts	39.123	0.2 μV	39.119
3 volts	39.087	20 μV	39.088

The 39 MHz link frequencies are in a 3:2 ratio with the 26 MHz radar frequency. This accident makes AM receivers most susceptible to interference from the link transmitters. Having even a low power link transmitter of this frequency range in close proximity to sensitive radar receivers is not ideal. This close proximity produced the most exasperating interference experienced during my development of AMOR. It affected the link from the Nutt site most severely. The AM receiver and FM transmitter are well isolated from each other electronically. Operating the whole system in a screened room produced no problems. In the field the interference was intermittent. When it was present checking the receiver video output indicated no problem. The mechanical layout of the remote site meant this was usually done after disconnecting the transmitter from the aerial and removing the box from its weather cover. Running the link without the main transmitter in operation suggested the FM data link was reliable. The problem was finally traced to an interplay of all four system elements at the Nutt site. Power from the link transmitter is radiated by the horizontally polarised link. Directly behind this link aerial the main receiver array picked this signal up and carried it to the AM receiver. When a strong 26.2 MHz signal was present a feedback loop was set up that produced the interference. The ground pulse from the 26.2 MHz transmitter array would always produce such a signal when running.[16]

The problem was solved by adding a 39 MHz trap to the AM receiver. Ironically the 3:2 ratio of the frequencies provided the solution as well as the root to the problem. An open ended $3\lambda/4$ length of coax cable was attached in parallel to the receiver input. Three quarters of a wavelength at 39.3 MHz produces a low impedence short at that frequency. This is precisely half a wavelength for the 26.2 MHz radar frequency. An open circuit half wavelength stub has a negligible effect on the signal.

The frequency chosen for the FM links is really too low. No commercial links could be found with a sufficiently large bandwidth to ensure an accurate

[16]In my experience on this project most interference problems are caused by your own equipment. The range of problems include: digital interfaces or computers too close to the receivers, faulty components, broken connections and the sort of feedback loops described here. For example, for short periods in the early morning Radio Australia might cause problems !

transmission of the amplitude data. The 39 MHz frequency is too close to the radar frequency for ease of development. The frequency is also prone to external interference. Data links operating in the UHF range near 450 MHz would be preferable. The technology to do this was not readily available when the links were originally constructed. Work has begun to upgrade the links using this frequency range.

2.5 Digital Hardware

The AMOR system uses digital hardware to record the radar data in computer memory. The 'Meteors Data Logger' is a high speed, five channel data gathering unit designed specifically for AMOR. It places the data directly into the RAM memory of an IBM-compatible AT computer. A plug-in interface card provides both input and output (I/O) and direct memory access (DMA) data transfers to the computer. Two external boards (built into one box) contain five parallel channels of A/D conversion with the necessary timing and multiplexing.

The five A/Ds make simultaneous conversions. The resulting data is multiplexed from buffers and sent to the computer. This transfer is done using the Direct Memory Access (DMA) controller in the computer. Every 2.64 ms an interrupt pulse is generated on the Timing Control board. This is sent to the computer interrupt controller. A software interrupt service routine called by the controller initialises a new starting location for the data, checks the previous sweep for meteors, synchronises data transfers to the extended memory and conducts a few other checks. The interrupt service routine used by the main observing routine is called Detector.

The hardware cycle repeats at 379 Hz with a new interrupt and range sweep. 480 μs after the interrupt pulse a series of 50 pulses at 40 μs intervals generates 50 sets of A/D conversions. A total of 250 bytes of data will be transferred via DMA to the computer. This process continues until all the allocated memory is full when the computer will start to overwrite the old data. The observing program `Observe.Pas` allows for 1800 data sweeps or just over 4.5 seconds of memory. The observing program needs to extract or *lift* the relevant meteor observation data from this raw data in this time. Alternatively the DMA transfers can be stopped and the data inspected at greater leisure. Many of the diagnostic test programs operate a 400 sweep data space for the DMA transfers.

2.5.1 A/D Conversion Board

This board consists of five separate analogue to digital conversion chips (A/Ds) and five track and hold chips (S/H for Sample and Hold) which digitise the analogue data. The A/D converters provide a linear 8 bit conversion for the 0 to 10 volt range of the analogue data.

The data from each A/D is buffer onto the corresponding latch which is then output to a common 8 bit bus which connects via a shielded ribbon cable to the DMA interface card in the MAC AT computer. The data is multiplexed into this bus by two counters (4017) the first of which allows the data to be read to the computer then clocks to select the next output latch. Five Data Taken pulses will come from the DMA interface card, one for each DMA read and these are used to advance the counter. The second counter is used to reset the DMA REQ latch and so stop the DMA read cycle after the fifth read. The next DMA REQ pulse to the computer will start the next set of five amplitude transfers.

The timing for an A/D cycle is all done from the Timing Control board. The DMA REQ pulse is generated when all five A/Ds (which are started in parallel) have completed their conversion. The Track and Hold chips are given a hold signal a microsecond or so before the Read pulse occurs to start a read cycle. The Read pulse is about 2 μs long. At the same time an LE (Latch Enable) pulse of about 1 μs moves the data into the latch buffers. The RD pulse now returns high which starts a new convert cycle. The time between successive read/start convert cycles is 40 μs.

The following signals on the A/D Conversion board can be checked for diagnostic purposes.

S/H, RD, LE, 4017 reset. All these come from the timing control board.

S/H latch reset. This pulse is generated at the end of an A/D conversion and goes to the timing control board.

DMA REQ. This line goes high after an A/D conversion and stays high until the next start conversion cycle.

Reset DMA REQ latch. This line goes low after the five data words have been read by the DMA controller. It goes to the DMA interface board.

Data Taken. This line has five pulses on it for every DMA REQ. It comes through the DMA interface card.

2.5.2 Timer Control Board

This board provides the timing and control for the five channel A/D board. The logic produces pulse trains (derived from a 800 kHz clock) with an overall cycle time of 2.64 ms. 66 pulses of 40 μs each are generated, but the first 16 of these pulses are gated off. This produces a gap of 480 μs then a train of 50 pulses at 40 μs intervals. After this the cycle repeats. These pulses initiate each A/D cycle which in turn starts a DMA cycle.

At the start of each cycle, a 40 μs monostable is triggered. This provides a trigger pulse to start the transmitter. An interrupt IRQ#3 is also generated to be sent to the computer. The TX trigger monostable can be turned on

or off from a front panel switch when the computer is not connected. This allows manual control of the transmitter trigger pulses. When the computer is connected the TX comes under software control via the I/O port of the DMA interface card. A receiver blanking monostable is also triggered from the TX monostable to allow blanking of the TX ground pulse in the receiver if required. Since no A/D samples are taken for about 400 μs after the transmitter ground pulse it was not necessary to implement any receiver blanking.

The following I/O control words can be sent to the timing control board. The control words are noted in hexadecimal.

$28	Disable all systems.
$30	Enable Timer Control board.
$F0	Enable complete system.
$F1	Enable system, with one of the I/O data lines high for a software timing test.

2.5.3 DMA Interface

This card plugs into the IBM-AT mother board, provides an eight bit input port, an eight bit output port and an eight bit DMA input channel. The input port has a memory address location of $310. The output port is located at $311. The location of the output port through the DMA interface card is defined by a constant, DMACard, in the OrbDefns.Pas unit[17]. DMACard:= $311. The card plugs into the IBM mother board and has ribbon cables coming from it to the external hardware.

The DMA channel consists of a 7474 DMA REQ latch which is clocked from the A/D card via a tri-state buffer. This latch is cleared by a pulse from the 4017 counter on the A/D board. On power up, the reset driver pulse from the computer resets the 7474 latch. A second latch blocks any REQ pulses until required. Output port bits 6 and 7 control these latches. DACK-3 anded with IOR provides the Data Taken pulse. A buffer is provided for the hardware interrupt.

2.5.4 DMA Controller

The IBM AT and XT range of computers both have DMA channels available for use by external hardware. The XT has one 8237 DMA controller for 8 bit data transfers. The AT has two, the second chip being slaved to the first to provide 16 bit transfers. In both cases channel 3 is available. DMA channel 0 is used by the computer for memory refresh. DMA channel 2 is used to control disk operations. To use a DMA channel a small amount of external hardware and some initialisation of the 8237 DMA controller chip is necessary. The

[17]The card was originally addressed as $301. DecnetDos required that its ethernet card have this address. The defined constant, DMACard, was introduced at that time. We have recently switched to using the file transfer protocol, 'ftp'.

hardware consists of a latch which is set by the requesting hardware. This latch output going high takes the DMA REQ line high which starts a DMA transfer. The 8237 issues a DACK (DMA Acknowledge) signal which can be used to reset the REQ latch. At the same time a Data Taken pulse is 'anded' with the DACK and this is used to open a tri-state buffer which puts the data on the bus. The 8237 will have set the address bus to the correct address for the data and provides the control signals to do the read or write.

If the DMA REQ latch is reset by a pulse other than the DACK then the REQ line can be held high allowing multiple DMA transfers. This is how the data logger of the AMOR system works. The REQ line is taken high and held high for the duration of five data transfers. The Data Taken line which pulses low for each transfer is used to multiplex the data onto the bus.

The use and initialisation of the DMA controller is fully covered by Eggebrecht (1987). The actual software statements used to do this are repeated below. The Page and MSB variables are preset address points for the next sweeps data. Briefly the required steps are outlined here.

1. Set up the starting address. This is a 16 bit address loaded into an 8 bit register; low byte first then the high byte to the same port (port $06 for channel 3). The final 4 bits (high bits) which make up the 20 bit address are loaded into a Page Register (port $82 for channel 3).

2. Set up a 16 bit Count Register to the number of bytes to be transferred. Load the low byte first then the high byte to the same port (port $07 for channel 3).

3. Set up the Mode Register to define the mode of DMA operation required.

4. Enable the relevant DMA channel.

```
Port[$82]:= Page;     {Load page register in DMA controller}
Port[$0C]:= $0;       {Reset byte pointer flip/flop}
Port[$06]:= $1;       {LSB of base address register}
Port[$06]:= MSB;      {MSB of base address register}
Port[$07]:= $F9;      {Word count register: LSB}
Port[$07]:= $00;      {                     MSB}
Port[$0B]:= $07;      {Mode, channel 3 demand write transfer}
Port[$0A]:= $03;      {Clear mask bit register for channel 3}
```

The system is now set up and will transfer data for each REQ until the Count Register reaches zero. The address register automatically clocks to the next address. Each page is 64 kBytes long. To reach the next page the Page Register must be incremented. If this is not done DMA transfers will continue but will overwrite data from the start of the same page. Care is taken within the observing program to ensure that individual data sweeps do not run over page boundaries.

The DMA controller services requests in a prioritised order. DMA#0 for memory refresh always has priority. The priority of the other three channels follows a descending order with increasing number. This priority order in the DMA controller can be reset if desired. It is essential that the A/D amplitude data transfer to memory has priority. As a precaution the program masks off both DMA#1 and #2 channels while echo amplitudes are being collected and file transfers to disk using DMA#2 are suspended. Every twenty minutes data acquisition is stopped and observation records written out to disk.

As noted above, it is possible to reset the DMA priorities and allow disk operations to proceed while the bus is not required for A/D data transfers but it is essential that file transfers to disk using DMA#2 do not cause radar data coming into memory on DMA#3 to be missed and implementing and testing this embellishment seemed an unnecessary complication. Disk operations also use the interrupt controller. As a deliberate policy all interrupts to the computer CPU, except the radar IRQ#3, are disabled during A/D data transfer operations. Writing data to disk takes a about three seconds every 20 minutes. Trying to do this concurrently would provide a negligible gain. [18]

DMA transfers take five processor clock times. An extra clock cycle is inserted as a wait state to ensure enough access time from memory and I/O ports. The CPU shares use of the bus with the DMA controller. The 8088 processor requires 4 clock times for bus transfers. With the computer operating at 12 MHz one clock time is approximately 83 nanoseconds. One byte can therefore be transferred every 0.83 μs. A memory refresh occurs every 72 clock times or 6.0 μs. In calculating the time required to transfer the five bytes from one A/D cycle into memory an extra DMA transfer for memory refresh should be included. DMA transfer for one 40 μs A/D cycle will take about 5.0 μs.

2.6 Decommissioned Equipment

As mentioned in the introduction to this chapter a considerable amount of equipment associated with radar the was replaced during the period of my PhD. The original conception of the radar was to handle signal processing for each of the receiver channels with a series of microprocessor boards that would produce a string of digital echo amplitudes when a meteor was present. These microprocessors were minimum system EPROMs with no I/O other than the 8-bit data lines. Development and debugging required modifying the 6502 machine language source code, compiling it and burning three new EPROM chips for each of the amplitude channels, transporting these to the field station, then

[18] I had originally envisaged using DMA#1 to do the data transfers into memory. This would have implicitly given the appropriate priority. During the detailed electronic design DMA#3 got adopted; I suspect for the rather arbitrary reason that IRQ#3 was being used as the interrupt channel for the AMOR system.

dismantling the signal processing rack, unplugging the appropriate board, replacing the chip, reassembling the rack and hoping the whole worked.[19] The code to run on each of the processors in these daisy chains was original developed in Pascal on the university mainframe computer. The switch to a PC based data acquisition system allowed the observing routines to be implemented directly in Pascal and provided for graphical memory dumps at each stage of the development.

The on board memory for each of the microprocessors in the daisy chain was only capable of storing a maximum of four sweeps of data. To make earliest possible detection of a meteor and ensure that this was reliable, a moving averager was included in the daisy chain. Considerable work was undertaken in trying to model and hence eliminate the worst effects this asymmetric filtering function had on data reduction. Almost all evidence of Fresnel diffraction patterns was smoothed out by its presence. The current computer has sufficient memory for the complete storage of 4 s radar data removing the need for an averager in the detection process.

The daisy chain microprocessors produced a one 8-bit output for each of the three amplitude and two phase channels when a meteor was present. These were assembled into an observation record by an Apple IIe computer and then written to magnetic tape. The Pertec tape drive used to record the observation data was designed for service with a mainframe computer as one of a several. Interfacing to the Apple IIe required a total of three digital boards and did not allow data to be read back at the field station. The tape drive worked well until it became necessary to transport the electronic components of AMOR to avoid them being stolen. The interface to the tape drive for example was stolen twice. The final decommissioning of the tape drive came when I traced a fault to one of the 1k $\times$ 1 memory chips (1103) in the on board buffer that even the manufacturer could not replace.

My decision to switch to a *luggable* MAC AT computer with an internal hard disk represents the most important engineering decision of my PhD. It reduced the number of custom built digital processor boards and interfaces from about 30 to 3 and replaced a 'read only' mainframe tape drive with a 40 MByte hard disk for data storage. All this came in a single easily transported case.

The original 26.2 MHz receiver antennas were constructed from coaxial cable joined core to shield every half wavelength to provide a broadside array. These did not have sufficient mechanical strength to support themselves and were susceptible to corrosion at the joints. Electrical breaks in the current open wire arrays can be easily traced by a simple visual inspection.

The original FM link receivers were modified versions of the 26.2 MHz AM radar receivers. They were replaced by receivers built around a commercial

[19] Often this was done in the very early hours of the morning whilst trying to get the thing running for a shower. I remember one memorable occasion accidently plugging the EPROM in back to front, wiping its memory with the 24 volt power rail and wondering why one channel was not working.

chip in an effort to improve sensitivity, cut down interference and stop a long term baseline drift in the output signal. The squelch circuit included in the chip helped improve reliability in using the link by causing it to drop out when the signal strength dropped below saturation. The baseline drift is probably related to temperature changes in the link transmitter circuits and was solved for the current configuration by a DC restoration circuit on the link outputs.

Chapter 3

Observation Software

Halfway through the period of my PhD research a decision was taken to employ an on site computer to identify and record the radar meteor observations.[1] This chapter describes the software used to do this. External hardware provides the timing and control signals necessary to place the echo amplitude data from the radio receivers into the computer memory. While the data for the current sweep is being placed into memory, the computer checks the previous radar sweep's data for a meteor echo. Where a positive detection is made the program lifts the necessary data set out of the memory and places it in an observation record. This is placed in a temporary store in the extended memory. Every twenty minutes the observation records are written out to the hard disk. At the end of an observation run the computer is brought back from the Birdlings Flat field station and the data transferred to the University main frame computer.[2] From here it is archived on magnetic tape. The data is then screened and reduced to produce heliocentric meteor orbits.

The meteor orbit radar system uses an MAC AT (IBM 286 compatible) computer to detect and extract meteor observations. This chapter describes the Turbo Pascal version 5.0 implementation of the observing program. A parent program is used to call the main observing program and to log the observing run. Data and timing signals from the receiver rack are feed into the computer using a custom built DMA interface card. The program makes specific use of various hardware features of the computer. The DMA controller brings raw data from the A/D converters into the computer memory. The IRQ controller initiates interrupt code to service the data from each sweep and provide a detection routine. The extended memory is used as temporary storage for observation records. The hard disk provides long term storage for the observation data. An overview of how the various components relate to each other is given in Figure 3.1.

[1] This decision to abandon the dedicated microprocessor system and switch to computer based data handling was the best engineering decision I made during my PhD.

[2] The 25 MBytes free on the hard disk will permit at least three days full recording.

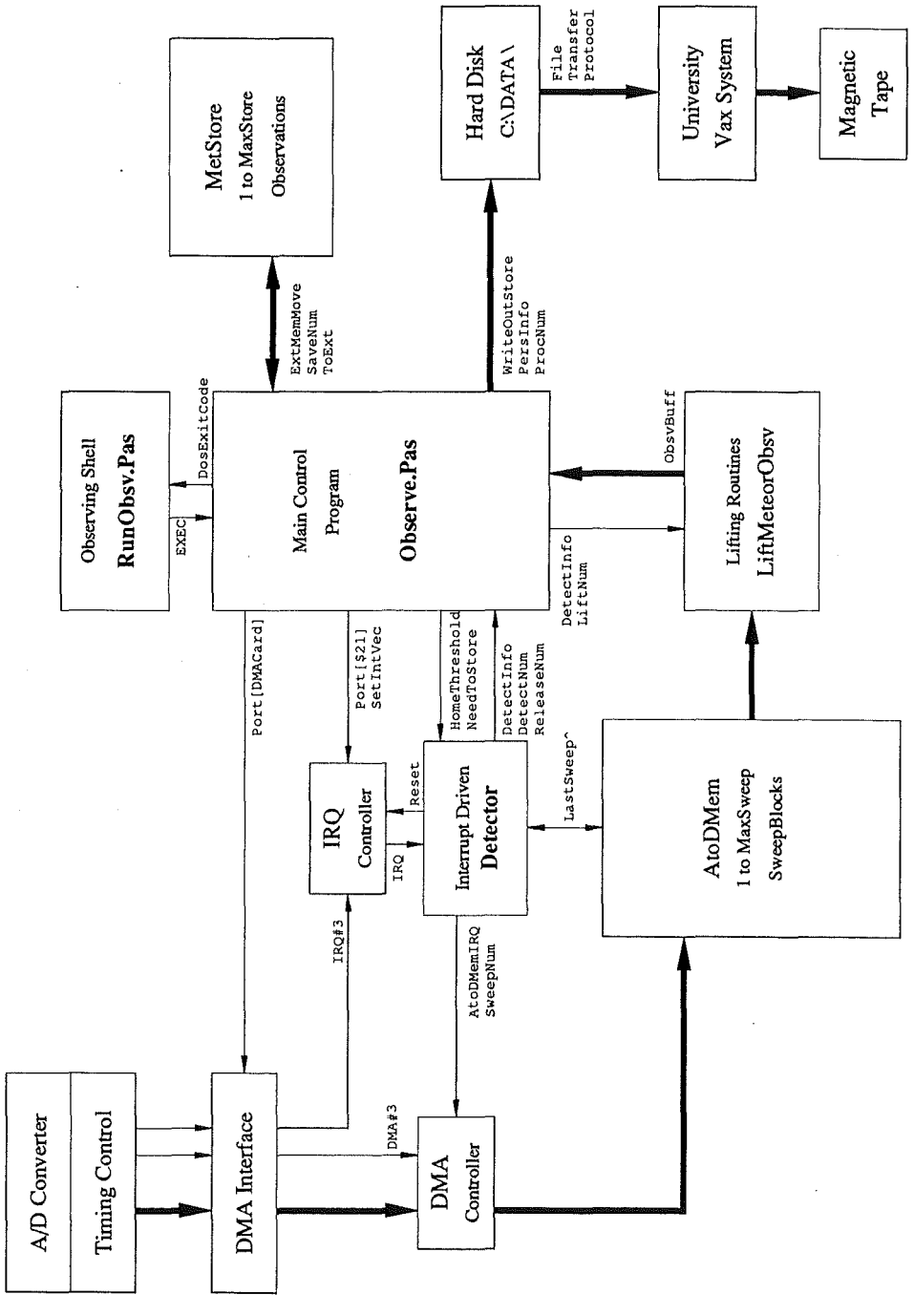


Figure 3.1: Diagrammatic layout of the main components in the observing program software. The main data flow follows the thick arrows. Control and information transfer is indicated by the thin lines. Relevant procedures and code units are in boldface.

The parent program, `Run_Obsv.Pas` provides a shell in which the observing program works. The parent program remains memory resident while the main observing program is run. An instruction to run the program is installed in the `AutoExec.Bat` file that ensures the program is restarted following a power failure. It ensures that when error conditions occurs within `Observe.Pas` the observing program is restarted. These include both expected error conditions within the code and runtime or I/O errors. The program gives a shell in which to fine tune how the system copes with unusually high noise, radio interference or saturation by sporadic-E sea echoes. This can then be done without touching the main observing software.

The main control program, `Observe.Pas` relies on a number of supporting code units. `OrbDefns` provides a definitions unit for the meteor orbit code. This includes declarations of variables and data structures used by both the observing routines and the reduction software. `OrbRadar` supports the raw data storage and detection routines. `OrbLift` contains routines to extract observation records, use the extended memory storage and write out observations to disk. `GenUtil` holds routines to read the hardware clock and format strings for display on the screen.

A number of test programs also use these supporting units. To test specific components within the observing program a number of diagnostic test programs are provided. In addition to the supporting code modules mentioned the graphics facilities in `OrbGraph` are also used. A description of these test programs is found in Appendix A. More details of the test routines can be found in Appendix B.

Every 2.64ms the Timer Controller Board sends a TX trigger pulse to the transmitter. It simultaneously sends an interrupt request on IRQ#3 to the computer. The interrupt controller within the computer uses the interrupt vector installed on software interrupt channel $0B to service this request. Interrupt $0B is set to point at the interrupt service routine detector by the main program `Observe.Pas`.

The interrupt driven meteor detection routine checks that no data overrun conditions apply within the AtoDMem space. It services the DMA controller to receive and store the incoming data of this sweep. The absolute addresses for storing the data are precalculated and available in the array AtoDMemIRQ. The new sweep data will overwrite the previous contents of the SweepBlock located at AtoDMem[SweepNum]^ within the AtoDMem.

The detector checks the echo amplitudes recorded by the Home site aerial during the previous sweep. If a meteor is identified the detector records essential information about the detection in the array DectectInfo. If the information previously stored in that record of DetectInfo has not been processed by the main control program then an overrun error occurs. The detector also decides whether the detection corresponds to a new meteor or to the redetection of an already existing echo. A new meteor is indicated by labelling it with the next consecutive storage number. A redetected meteor is labelled

with the number of the previous detection. This information is also passed to the controlling program in DetectInfo.

The control program, Observe.Pas waits until the detector releases a positive meteor detection to it. If the meteor trail was previously detected the time is added into the PersInfo array for that meteor. For a new meteor detection the routine LiftMeteorObsv is called to collect the observation data into the observation record ObsvBuff. The contents of ObsvBuff are then transfered up to the extended memory MetStore. The BIOS Block Move service that does this transfer requires that interrupts be masked out during the transfer. To ensure that none of the start of sweep IRQ#3's is missed during this protected operation the transfers to MetStore are sychronised by the interrupt driven detector. NeedToStore is set true and the detector does the transfer during the next convenient sweep.

When a Home site detection has been made the control program uses LiftMeteorObsv to transfer the relevant data into the observation buffer ObsvBuff. It looks for a positive meteor detection in the data recorded by the Nutt and Spit site receivers. Where no echo is detected an amplitude sample from that channel is recorded. A detected echo is lifted and stored in the SiteProfile vector. The phase data in the same range bin as the Home site detection is lifted into the Tin and Tos vectors. An average noise sample for each of the sites is taken from other range bins in the sweeps around the sweep the meteor was detected in. If the noise is too high, 255 is returned and the control program terminates the run with Halt(2).

The observing code detects, lifts and stores 240 meteor observations within a 'storage' run. Once this is done the controlling program writes the observations in MetStore out to disk. The routine WriteOutStore selects the appropiate data file and appends the observation to it. It adds the duration to the record if the echoes from the meteor trail lasted for longer than half a second. This duration of persistence is recorded as a single byte in the observation record. If the trail lasts longer than $4\frac{1}{4}$ minutes its duration is recorded as 255 seconds.[3] The observation record is then written to disk. Once the contents of MetStore have been written out the controlling program resets its control variables, zeroes the detection array and any DMA hold flags. This is done with the routine ResetVars from the `OrbRadar` unit. It then starts another run. The observer has the option to dump the current contents of MetStore to disk and continue with a new storage run. This is done by pressing any key. To terminate the program completely, press S.

Various conditions occur which require some observer intervention. These generate an alarm and terminate the observing program. The steps necessary to ensure an orderly shutdown are contained in the exit procedures OrbitExit and LiftExit. These are installed by the initialisation sections of the mod-

[3] A one byte variable in the observation record is used to record the duration of a persistent meteor. This makes it easy to record echoes from 0 to 254 seconds and indicate an overflow with 255.

ules `OrbRadar` and `OrbLift` respectively. Alarm conditions occurring within the interrupt driven detection routine terminate the observing program with Halt(1). Halt(2) is used to terminate the program for any unusual conditions occurring within the main control program `Observe.Pas`. Any errors occurring within the routines that control data movement to and from the extended memory use Halt(3). A runtime error is handled in the same way. In this case the ExitCode is zero but ErrorAddr is set to point at the memory location of the error.

3.1 Memory Use and Data Structures

The meteor orbit observing program is implemented in Turbo Pascal. Figure 3.2 provides a layout of memory use by this program. It shows the relative arrangement of the code and data segments, the stack, the AtoDMem space with its alignment buffer on the heap and the extended memory MetStore. For more details on memory use by Turbo Pascal see the Memory Map (Figure 15:1) in the Turbo Pascal Reference Guide.

The computer installs DOS version 3.30 and DosEdit at startup. These are contained in the operating system box on the memory map and require about 68 KBytes in total. A compiled version of the `Run_Obsv` is then automatically executed. The observing program can be run from DOS in its compiled form. Copies of both the executable images `Run_Obsv` and `Observe.Exe` are stored on disk in the C:\OBSERVE directory. When either is run the operating system builds a 256 byte Program Segment Prefix and then loads the executable image.

Above the programs, memory space is allocated to the stack. This provides temporary storage for variables when calling procedures. The size of the stack can be modified using the {$M} compiler directive within the source code. I have tried setting the stack to a maximum size of 4 KBytes. This caused a runtime error 202 (stack overflow) whilst writing out records to disk. It seemed easiest to leave the stack at its default size of 16 KBytes.

The available memory above the stack is called the heap. The memory is allocated dynamically by the heap manager at run time. Heap use within `Observe.Pas` ensures that there is never any free memory below the heap pointer. The free list will therefore be empty and the free pointer directed to the top of DOS memory. Overlay code is not used so there is no overlay buffer. The heap is used to provide temporary storage for the raw amplitude data. The array AtoDMem contains pointers to each sweep block of data.

The extended memory is used to provide temporary storage of observation data records. It is set up as an array of address locations at the start of the observing program. The actual contents remain undefined until observation records are written up to it during a storage run. Trying to read the contents of the MetStore before they have been written into will probably cause a parity

error.

The observing program Observe.Pas uses most of the memory available within the computer. The free space in the extended memory is available for use if the system is upgraded to provide data reduction while it is collecting the raw observation data; see section 3.1.4. The extra space in the DOS accessible memory allows some freedom in the program code size. For simplicity the observing program does not include any graphical display of the observation records being collected. It might become convenient to do this. The DisplayObsv routine requires an extra 25 KBytes of memory to include in the program. Future upgrades of the DOS operating system will inevitably eat away at this free memory. The actual size of the unused DOS accessible memory is easily changed by modifing the number of sweeps of data stored in the AtoDMem. Section 3.1.2 provides some details for doing this.

3.1.1 SweepBlock Record Structure

The receiver video and phase comparison amplitudes are digitised by the A/D conversion board. The raw data is then transfered directly into the computers DOS accessible memory by the DMA controller. The information collected during a sweep is stored in a Pascal record of type SweepBlock. This is a 256 point array of 8-bit amplitudes. Figure 3.3 depicts the layout of the record structure diagramatically.

The first byte of data from the A/D board is transfered into index location $i = 1$. In each sweep 250 bytes are transfered so the last byte is stored at index location $i = 250$. The first byte of the SweepBlock record, $i = 0$, is reserved for use as a DMA hold flag. The DMA hold flag contains zero unless it has been set (to 255) by the detector. Data are stored sequentially in the same order that it is sent to the computer. Every $40\mu s$ another set of five samples is taken then transferred. The amplitudes are sent in the order: Home, Nutt, Spit, Tin, Tos. Each group of five bytes constitutes the data from one range bin. The last five bytes of the record are not used and should always remain preset to zero. These are included in the sweep block to ensure that page alignments are preserved to ensure sensible behaviour of the DMA controller.

The detection routine also uses the SweepBlock record structure. It uses an array DetectCount to store information for the detector from one sweep to the next. The index locations where the information is recorded indicates the site and range bin to which it applies.

The DMA controller is reset at the start of each sweep to handle the data transfer of that sweep. It is run in the demand write transfer mode. Each time a byte is transferred the target address is incremented by one. The 4-bit page register however can only be changed by resetting the controller. The contents of a sweep block therefore must never cross a 65 KByte page boundary in the DOS memory.

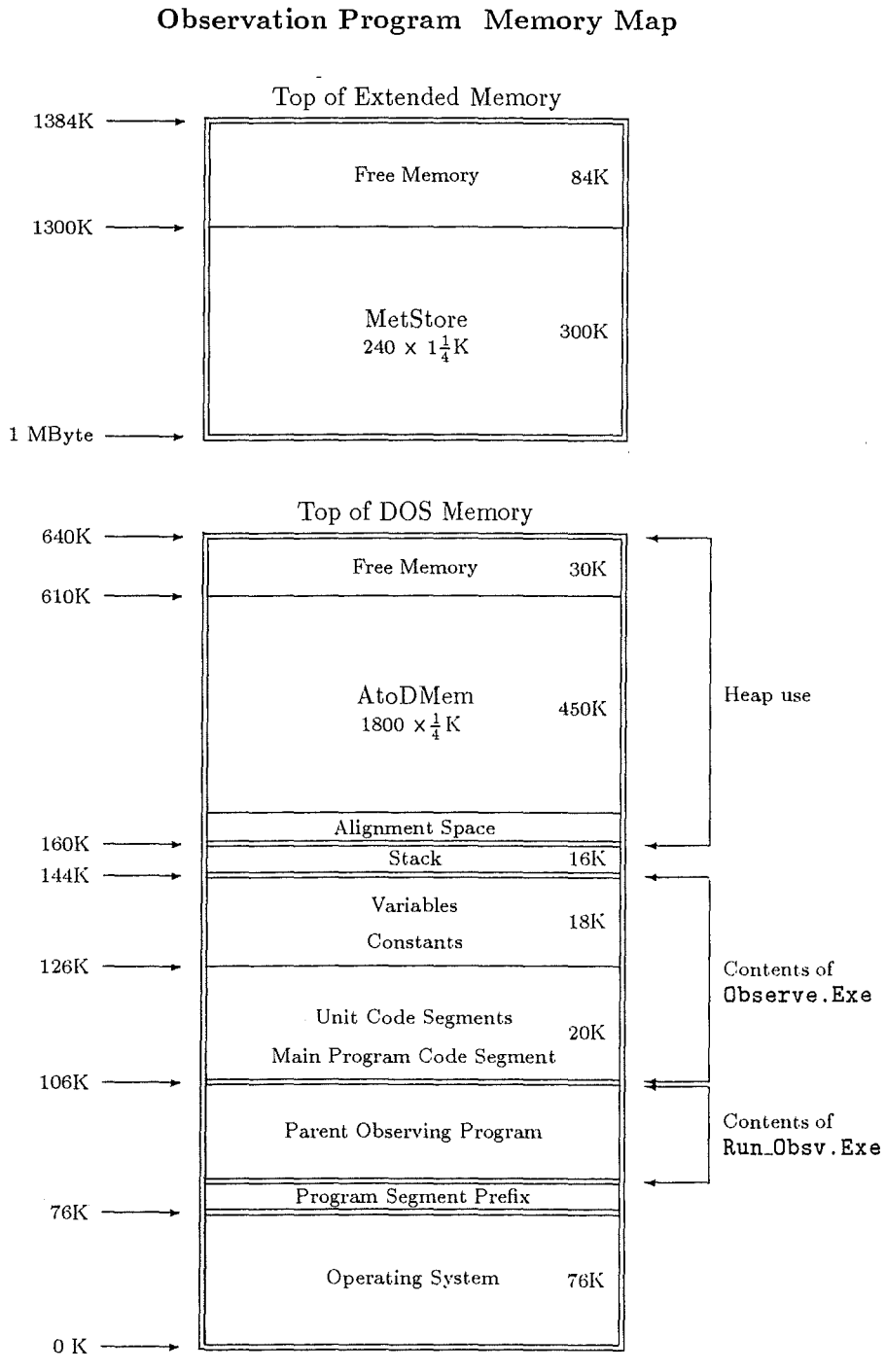


Figure 3.2: Memory map for the Turbo Pascal implementation of the meteor orbit observing program (not to scale). It shows the relative arrangement of the code and data segments, the stack, the AtoDMem space with its alignment buffer on the heap and the extended memory MetStore.

SweepBlock data array

0 DMA	1 Home	2 Spit	3 Nutt	4 Tin	5 Tos
	6	7	8	9	10
	11	12	13	14	15
	.	.	.	.	.
	.	.	.	.	.
	.	.	.	.	.
	246	247	248	249	250
	251	252	253	254	255

Dummy Data (RB#65 of last sweep)

Range Bin #16

Range Bin #17

Range Bin #64

Not used (preset to zero)

Figure 3.3: The digitised data collected during one sweep is stored in memory as a Pascal record of type SweepBlock. It is a 256 point array of 8-bit amplitudes. The index offsets are in the top left of each box.

3.1.2 AtoD Memory Space

The AtoD memory space is a 450 KByte area reserved as a temporary store for the digitised raw data. It comprises a consecutive series of sweep blocks numbered from 1 to MaxSweep. The pointers to each of these sweep blocks are stored in the array AtoDMem. To simplify and speed up the process of resetting the DMA controller each sweep the absolute page and MSB addresses of the blocks are precalculated and stored in the array AtoDMemIRQ.

The procedure SetAtoDMemory from the unit `OrbRadar` sets up this memory at the start of the observing program. The Turbo Pascal heap manager allocates memory for each successive sweep block on the heap. The sweep blocks of the AtoDMem are allocated once at the start and never released. This constitutes the only use of dynamic memory by the observing program. It greatly simplifies heap use.

To ensure that a sweep block never crosses a 65 KByte page boundary the program uses an alignment buffer. This is a section of unused memory between the top of the stack and the first sweep block of data. The variable HeapPtr which points to the base of the heap is set to point to the first free multiple of 256 bytes. The segment address is therefore a multiple of 16. The program SetAtoD.Pas can be used to check that the memory locations of the AtoDMem sweep blocks are correctly aligned.

The AtoD memory space usually uses one of two sizes. When running the observing program the AtoDMem contains FieldMax = 1800 sweepblocks. This corresponds to the 450 KByte size. During a storage run the contents of the AtoDMem are overwritten every 4 to 5 seconds.

To facilitate running programs while in the interactive Turbo environment MaxSweep is set to TestMax = 400 sweep blocks. This allows about a seconds worth of data to be stored in the AtoDMem. Most of the diagnostic programs use this option. MaxSweep can easily be set to anything less than FieldMax. To use a larger AtoD memory space the array declarations of AtoDMem and AtoDMemIRQ would need to be expanded.

The contents of the AtoDMem can be inspected using LookAtSweepBlocks. This provides an interactive environment to look at the data contained in any sweep block. The LookAtDetectCount procedure has the same basic layout as the display for each sweep block.

3.1.3 Observation Record

Each time a meteor is detected the handling program lifts the relevant data into an observation record. This is the basic archiving record for the raw observation data. The Pascal record contains a total of 1280 bytes of binary data.

The date and New Zealand Standard Time, NZST of the meteor event are contained in the first six bytes of the record. For consistent data records the CMOS clock in the computer should always be set to NZST. It should not be

reset to New Zealand Daylight Time during the summer. The meteor number is the sequential detection number within a storage run. It also corresponds to the storage location in the extended memory MetStore that held the record until it was written to disk. PersTime is the duration, in seconds, of the meteor echo. Ionisation trails which last longer than 254 seconds are recorded as having lasted 255 seconds. The DetectionSite indicates the receiver channel on which the meteor was detected. DetectionSite = 1 implies Home site detection. Some earlier versions of the observing program ran detection routines on the Nutt and Spit sites as well. I am reasonably happy with running the interrupt detection routine on just the Home site. The observation record could be modified to always imply this. The byte, DetectionSite could be used by PersTime to extend the recorded meteor duration past the current four minute barrier. The noise samples N1 (Home), N2 (Nutt) and N3 (Spit) are average values from other range bins around the sweep the meteor was detected in. The first 40 or so amplitudes of the site profiles provide a noise sample from the relevant range bin before the echo rose above the detection threshold.

Each of the three receiver stations stores information about the echo amplitudes in a SiteProfile record. IndexOne is the sweep that the data in the first byte of the amplitude vector come from. DetectSweep is the sweep number a meteor echo was detected in. IndexOne should be 49 sweeps before DetectSweep. If no meteor was detected on that channel then DetectSweep should be equal to IndexOne. RangeBin indicates which 40μs bin the meteor echo occurred in. Threshold notes the detection threshold being used when the meteor was detected. The 250 point array, Vector contains the echo amplitudes for successive sweeps. The vector array is therefore a time series of echo amplitude sampled at 2.64ms intervals. These are quarter, half, quarter averages centred on the detection range bin, RangeBin. The Tin and Tos vectors contain the phase comparison amplitudes from the two Home site aerials. They are a direct lift of the phase channel contents at the same range bin as the Home site RangeBin.

The contents of an observation record can be displayed graphically on the screen using the DisplayObsv procedure. The code unit OrbGraph contains the routines necessary to do this. A menu driven routine to look at observation records stored on the hard disk is available in the unit OrbView. The main procedure DisplayDiskObsv can be run from the program LookDisk.Pas.

While in the interactive Turbo environment the debug watch can be used to view the numerical contents of any observation record in say ObsvBuff. In combination with the record layout of Figure 3.4 this is very straight forward. Once the record has been read into ObsvBuff it should be entered as a watch, cntr-F7. The first 12 values in the watch window correspond to the first line of Figure 3.4. Placing a watch on ObsvBuff.Home for example displays the second line of Figure 3.4.

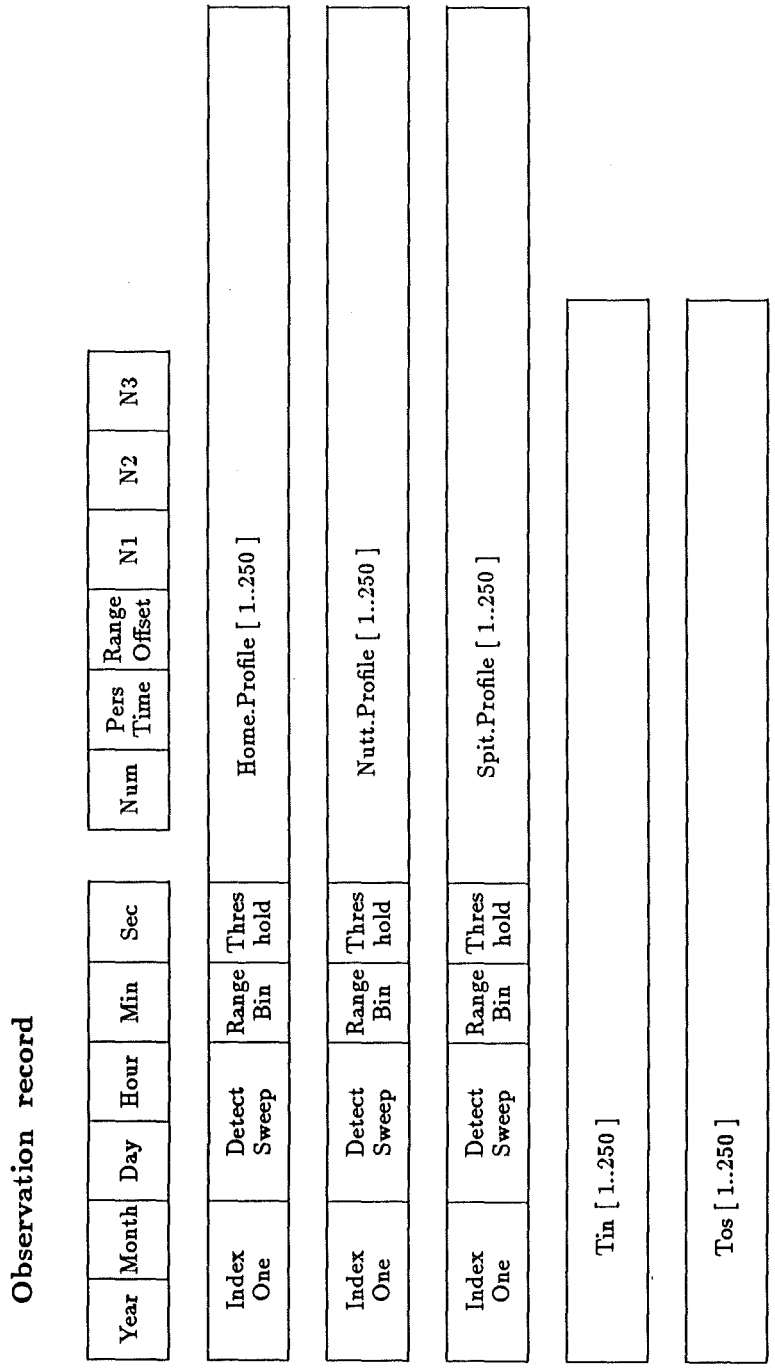


Figure 3.4: The raw observation data are stored in a Pascal record of type Observation. It is a 1280 byte record of binary data. Each square box in the diagram represents one byte. The 250 point vector boxes are not to scale.

3.1.4 Extended Memory Store

The standard configuration of the MAC AT computer comes with 384K of extended memory. This is used by the observing program to provide temporary storage for observation records during a storage run. A 300 KByte section of this is used as an extended memory store for meteor observation records, called MetStore for convenience. The start address locations for the records in the store are available in the array MetStore.

Writing records directly to disk is not possible while the detection routines are being run. IRQ#0 is required to service the disk drive. The actual transfer of data to the disk is done by the DMA controller on channel 2. These DMA transfers to disk have a higher priority than the data transfers into the computer on channel 3. Both the interrupt processing and the DMA transfers take time and would cause sweeps of data to be missed. Data transfers to and from the extended memory are controlled by the central processor. The 1280 byte transfer of an observation record can be comfortably fitted in between radar pulses.

The MetStore is organised to provide temporary storage for 240 observation records. The variables which provide the numbering of meteor events are declared as bytes. This restriction makes it relatively straight forward to modify the number of records stored to anything less than 255. Setting MaxStore = 240 is a convenient round number less than this upper limit.

If a decision is made to modify the orbit radar to run completely unattended for days at a time some data reduction of the observation records would be advantageous. This is especially true where interference and sporadic-E echoes trigger false detections create large numbers of bogus observation records wasting space on the hard disk. The updated version of Observe.Pas could do the extra processing in the quiet time between meteor events. The extra room in the extended memory is then available to be used as a temporary store for the ReducedData records.

3.2 Interrupt Driven Detection Routine

This section gives a more detailed description of what the meteor detection routine does after each transmitter pulse. It is implemented as a Turbo Pascal interrupt procedure. The routine is initiated by an interrupt generated at the same time as the TX trigger pulse. The pascal procedure is called Detector and is found in the code unit `OrbRadar`. A pointer to the detector routine is installed on channel $0B in the interrupt vector table.

The procedure uses the Turbo Pascal interrupt directive which ensures that all the registers are automatically saved in the entry code of the routine. It does not re-enable further interrupts. The hardware interrupt controller is reset at the end of detector and interrupts to the CPU are re-enabled with an STI instruction using an inline statement. The exit code restores all the registers

unmodified and executes an interrupt-return instruction. For more specific details on interrupt service routines in Turbo Pascal refer to the Reference Guide p221.

If the procedure takes too long to execute then the next interrupt will be missed. It will arrive before the interrupt controller is reset. A missed interrupt means the data from that sweep is lost. This would cause an error in the time lags calculated from the profiles of up to one pulse for each sweep missed. The current inplementation of the detection routine with range checking disabled will handle up to six simultaneous meteor detections. That is there is enough time in one sweep to record detection information about six echoes and still finish before the next IRQ#3. This number of detections is statistically most unlikely under normal observing conditions.[4] Any occurrences of this are assumed to be caused by a transient noise spike or a raised baseline voltage. They cause the program to terminate with an error condition.

The computer normally runs a system clock and disk drive monitoring routines off the hardware timer interrupt IRQ#0. It is not always possible to execute all this code and the detection code in the time between successive IRQ#3's. This also causes a loss of interrupts. When the observing program enables IRQ#3 to the interrupt controller it also masks IRQ#0. This prevents the intermittent loss of sweeps. It is necessary to re-enable these interrupts for orderly disk use while writing out observation records. Memory refresh is done by DMA channel 0 and does not require any processing associated with IRQ#0.

If any other interrupt services are added to the observing code considerable care should be taken to ensure that no IRQ#3's are missed. Most of the various interrupt test routines maintain a long integer variable, IRQCounter that is incremented each time the code is run. The number of hardware interrupts sent to the computer can be counted by a frequency counter. Comparing these two numbers provides a check that no interrupts have been missed.

The positive detection of a meteor requires that the receiver amplitude at a particular range bin remains above threshold for six consecutive sweeps. The threshold value can be set fairly well above noise. I have avoided the added complexity of having it adjustable during a run. The level is set at an A/D amplitude of 30 in the observing program. Since data from the leading edge are stored in memory it is possible to leave the final decision about the positive detection of a meteor until well into the event. The number of consecutive sweeps above threshold should not be set too large. This will miss short duration and especially lower amplitude echoes. If the number is too small the detector tends to trigger off spurious noise spikes. These sometimes last for two or three sweeps. Six sweeps above threshold was found to be an optimum meteor detection criterion.

In any particular storage run the detector keeps track of which meteor is

[4] Virtually impossible

being referred to by several index numbers. SweepNum is the reference number for the sweep block of data within the AtoDMem space being collected during this sweep. It is initialised as zero before a storage run starts and wraps around from MaxSweep back to one. The detection routine processes the amplitude data collected in the last sweep. NewNum is a sequential counter that is incremented each time there is a new meteor trail detected. The storage location within the MetStore, SaveNum, to which the observation record is later transferred is set equal to NewNum. DetectNum is an index location within DetectInfo. Each time a new detection is made DetectNum is incremented. Detection information is written into the DetectInfo[DetectNum] record. It wraps around from MaxDetect back to one. ReleaseNum points to the last meteor detection that was released by the detector. This allows the handling program to begin processing the detection. ReleaseNum follows DetectNum through the DetectInfo array but about half a second (200 sweeps) behind. LiftNum is also an index within DetectInfo. It points to the detection infomation currently being used by the main control program. As such it follows ReleaseNum through the DetectInfo array. The difference between the two numbers depends on the meteor echo rate at the time. LiftNum is *not* modified by the detection routine. If DetectNum ever catches up to LiftNum then detection information is being overwritten before the lifting routines have finished processing the detection. This causes the program to terminate with Halt(1).

The array DetectCount stores information for the detector from one sweep to the next. It has the same SweepBlock structure as the A/D data in the AtoDMem. Figure 3.3 gives a pictorial view of its layout. While the detector is looking for a meteor, DetectCount provides a count of the number of consecutive sweeps that the Home site echo amplitude in a given rangebin has been above the threshold. Once a meteor has been detected, DetectCount[i] = 6, the number is used as a counter to ensure another 200 sweeps are collected before the detection is released for further processing. When DetectCount contains 255 that range bin has been temporarily screened off from the detector algorithm indicating that a meteor has been detected in a neighbouring range bin. DetectCount also contains information about the last meteor that was detected at that range bin. This is stored in the Nutt, Tin and Tos columns at the same range bin and is describe further below.

Whenever a new meteor echo is detected a DMA hold flag is set approximately 100 sweeps prior to the detection sweep. This ensures that the data in the AtoDMem are not overwritten before the handling program has lifted the observation record. If the DMA hold flag in the next AtoDMem block is set, AtoDMem[SweepNum]^[0] = 255, then the program is terminated using Halt(1) as a data overrun error. If two meteor echoes are detected in the same sweep then the detection routine moves the location of the flag a little to ensure that each detection has a unique DMA hold flag.

Information about a meteor detection is passed from the detector to the

handling program and lifting routines in an array DetectInfo. Each entry contains the information in a record of type DetectRecord. The layout of this record is displayed in Figure 3.5. DetectInfo contains MaxDetect = 48 detectRcord records. This provides enough room for the handling program to process six simultaneous persistent echoes without having the detector overrun the detection information before it is used. If an overrun condition occurs the program is terminated with Halt(1). The contents of a DetectRecord can be viewed using the procedure LookAtDetectInfo in the `OrbRadar` unit.

When a meteor echo is detected the time of the event needs to be recorded. Since the system clock is disabled the detection routine needs to read the time off the CMOS hardware clock. This is done using the BIOS read real time clock; interrupt $1A, subfunction $02. The hardware clock can be reset by running the set up routine while rebooting the system. It should always be set to New Zealand Standard Time. A number of routines within the `GenUtil` unit use the BIOS interrupt $1A, Time of Day services to read the date and time off the hardware clock. The procedure Timestr can be used to format the time into a string for display on the screen.

When a positive detection occurs the routine checks to see if it is a new meteor trail or a redetected persistent one. When a new meteor is detected its associated label, NewNum, is noted at the same range bin in the Tin column of the DetectCount array. The time at which the detection occurred is recorded in the number of seconds since the start of the current storage run. To allow for some fluctuations in the detected range of a meteor, due to both radio noise and motion of the target trail, the information about the last detection is also stored in adjacent range bins on either side of the detection.

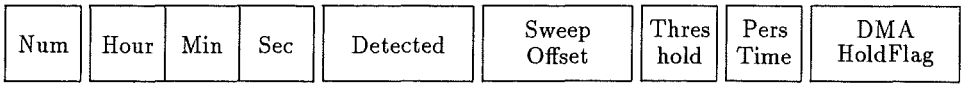
If an echo is detected within two seconds at the same or neighbouring range bins it is assumed to be a redetection of the previous meteor trail. The time of the latest detection is updated within DetectCount. The number of the original detection and the time since the most recent detection are noted in DetectInfo. In this case no DMA hold flag is set. Only the profiles collected during the initial formation of the trail are useful for orbit calculations. A redetection will not occur until after the active screen is cleared and the original has been released to the lifting routines. It is however possible that the original detection has not been processed by the lifting routines.

If no meteors have been detected in the last five minutes then the program is terminated with Halt(1). Such a long interval between meteors is highly improbable and the condition is used to indicate that the transmitter has stopped.[5]

The program TestDete.Pas is a demonstration and test bed for the detection routine Detector. It loads the AtoDMem with sweep data from one of the test files < name >.Dmp in the C:\DATA directory. The detection routine is

[5]Of course if no observer is present then the `Run_Obsv.Pas` program restarts the program in case it was just a long interval between meteors.

DetectRecord record



Number	The save number associated with the detection. For a new detection it corresponds to the next SaveNum location in the MetStore. For a redetected echo it is the SaveNum number of the original detection.
Hour, Min, Sec	New Zealand Standard Time of the meteor detection. The date of the observation is added when the observation record is written out to disk. This is read from the CMOS clock using the BIOS interrupt $1A service.
Detected	The sweep number in the AtoDMem that the meteor was detected in. This corresponds to the sweep in which the echo amplitude has been above threshold for six consecutive sweeps.
SweepOffset	Index location within the SweepBlock array of data at which the positive detection was made. This location implies the value of the rangebin for the meteor echo.
Threshold	Home site threshold A/D level being used by the interrupt detection routine.
PersTime	Time in seconds since the last detection of a long lasting echo. If this is a new echo the number PersTime will be zero. It is also possible that less than a second has elapsed since the previous detection for a persistent meteor trail. In this case Perstime would also be zero.
DMAHoldFlag	The sweep number within the AtoDMem that contains the DMA hold flag corresponding to this meteor detection. This will usually be 100 less than the detection sweep Detected. Where two or more meteor echoes are detected in the same sweep the location of the hold flag is adjusted so that each detection has a unique flag.

Figure 3.5: The detection routine records information about a detection in a DetectRecord record. This is referred to as DetectInfo[DetectNum]. Both the interactive debugger and the routine LookAtDetectInfo display the information in this order on the screen.

called by a software interrupt to process each successive sweep. The program runs through the data in several different ways to highlight the critical points in the detection process. Details of these are in the test program description section.

TestDete.Pas does not provide a rigorous check on whether the detection routine is fast enough. The routine is not being run off an externally generated hardware interrupt so comparing the number IRQCounter with the number of interrupt generated would not work. More seriously no DMA transfers are being done while the interrupt code is being processed. The processor does not have to wait the necessary clock cycles while the transfers are being done. Since the program does not have this time critical aspect it does provide a test bed for any changes since it can be run with the range checking switched on.

The various operations within the detector are described in the list below.

Interrupt Entry Code The contents of the registers are saved to the stack. Other interrupts are not re-enabled.

Update variables

- SweepNum is incremented from the last sweep collected. It cycles from one through to MaxSweep.
- The pointer to the last sweep block collected, LastSweep is found from the array AtoDMem. On the first sweep of a storage run the detector looks at the contents of sweep number MaxSweep. This is initialised to contain zero.

Reset the DMA controller

- If the DMA hold flag for the next sweep is set then the run is terminated with a Halt(1). This ensures that echo amplitudes within the AtoDMem are not overwritten before the lifting routines have finished with them.
- The DMA controller is loaded with the absolute address locations for this sweep. These are precalculated and available in the array AtoDMemIRQ. The data always starts at an offset of 1 in the sweep block leaving the DMA hold flag as is. The controller is set to accept 250 one byte transfers before needing to be reset. The controller is set to the demand write transfer mode. Finally the channel 3 mask bit is cleared so the transfers can proceed.

The program then looks at the amplitude data from the Home site receiver. Information about the recent echo amplitudes is stored in the array Detect-Count. For each range bin #18 to #60 the following is completed. That is from LastSweep^[16] to LastSweep^[226] in steps of 5 do the following.

Detector Where DetectCount[i] is less than 6 the routine is looking for a meteor. Compare the signal level from LastSweep^[i] with the detection threshold HomeThreshold.

- If signal < threshold then reset DetectCount[i] to zero.
  - No detectable meteor present.
- If signal ≥ threshold then increment DetectCount[i].
  - If DetectCount[i] = 6 then a positive meteor detection.
  - If DetectCount[i] < 6 then a possible detection.

Meteor Detected DetectCount[i] = 6.

- Increment DetectNum and store the detection information in the record DetectInfo[DetectNum]. Check there is no overrun within the DetectInfo array.
- Check to see if a meteor echo has been detected in a nearby range in the last 2 seconds. If so this detection is assumed to be the continuation of a persistent echo.
- If the meteor echo represents a new detection then set a DMA hold flag. This will stop the DMA data acquisition at the SweepBlock 50 before the current 'detection' sweep. It is cleared by the control program once the meteor observation has been lifted.
- Blank adjacent range bins with a meteor active screen. That is

  $$\begin{aligned} \text{DetectCount}[\, i-15, i-10, i-5\,] &= 255 \\ \text{DetectCount}[\, i\,] &= 7 \\ \text{DetectCount}[\, i+5, i+10, i+15\,] &= 255 \end{aligned}$$

  Note: For range bin #60, i+15 corresponds to i = 241. For range bin #18, i-15 corresponds to i = 1.
- Record the meteor number associated with this detection and the time of detection in DetectCount. The time comes from the storage run second counter PersCount.

  $$\begin{aligned} \text{DetectCount}[\, i+2\,] &= \text{High byte of PersCount} \\ \text{DetectCount}[\, i+3\,] &= \text{Low byte of PersCount counter} \\ \text{DetectCount}[\, i+4\,] &= \text{Number of original detection.} \end{aligned}$$

  These are also transferred to the adjacent range bins both two closer and two further away.

Meteor Running if DetectCount[i] is greater than 6 then a meteor has been detected in that or a neighbouring range bin in the last half second.

- DetectCount[i] = 7 to 205 indicates the meteor profiles are still being collected. Increment DetectCount[i] by one.
- If DetectCount[i] = 206 then half a second of data has been collected since the meteor was detected.

  - Increment the variable ReleaseNum to tell the main control program that the meteor profile has been collected. ReleaseNum will be the index that points at the relevant information for that meteor within DetectInfo.
  - Remove the meteor active screen; that is, set the adjacent range bins where DetectCount[i] contains 255 to equal 0. The routine checks to see if there are any active neighbours that should leave a screen on one of these range bins within DetectCount.
- DetectCount[i] = 255. The range bin is screened to avoid redetecting a meteor whose amplitudes spill over into adjacent range bins as they evolve. The routine should continue with the next range bin.

Re-enable interrupts to both the IRQ controller and the CPU.

Move to MetStore If the main control program has set NeedToStore true then the contents of ObsvBuff are shifted to MetStore. If the routine has spent time dealing with the detection or release of a meteor then the transfer is postponed until the next sweep.

Exit Code Restores the registers and executes a return from interrupt.

3.3 Lifting an Observation Record

Once a meteor echo has been detected another half second's worth of data is collected. The detector then releases the detection for further processing. At this point all the raw data necessary to calculate an orbit are resident in the temporary storage space AtoDMem; buried in a deluge of raw digitised data. An observation record provides sufficient information to calculate a meteor orbit. The routines to 'lift' the data for one meteor observation out of the AtoDMem are found in the OrbLift unit. The procedure LiftMeteorObsv controls the assembly of the observation record into the buffer ObsvBuff.

The detector places information about a meteor detection into the DetectInfo array. The main control program Observe.Pas decides which meteor detection is to be lifted. It uses LiftNum to identify the record within DetectInfo that applies to the detection. The time of observation is transferred directly to the observation buffer ObsvBuff.

Since the meteor echo was detected on the Home site channel both the range bin and sweep it occurred in are known. It is therefore a straight forward matter to lift the Home site amplitude profile. The procedure LiftProfile starts at a sweep 49 before the detection and lifts a quarter, half, quarter amplitude average from the three range bins centred on the detection index DetectInfo[LiftNum].SweepOffset.

A noise sample of the three sites is taken. This is an average value taken from other range bins around the sweep where the detection was made. If less

than 40% of the values sampled are below the 30 A/D units threshold a noise value of 245 is returned. The detection routines would be unable to operate properly under these conditions. An alarm is triggered by the main controlling program if this occurs.

LiftMeteorObsv then runs its own detection routine on the data recorded from each of the Nutt and Spit sites. The function FindProfile uses the same detection criterion as the primary interrupt detector. A detection threshold 10 A/D units above the noise level for that site is used. Six consecutive sweeps with the amplitude above the detection threshold in that range bin give a positive detection. FindProfile starts looking at a sweep 100 before the home site detection. This corresponds to the location of the DMA hold flag. Timelags of over 100 pulses are very unlikely to occur. The station layout and geometry dictate that the range to the ionisation trail will be approximately the same for all three sites. The location of the current receiver stations and the delays caused by transmission across the links mean differences of up to 10 or 12 km are possible. Not all the range bins need be searched. Time delays associated with the communication links from the remote sites cause data from the same range to be recorded at larger range bins. The FindProfile detection routine searches the five range bins larger than the Home site detection RangeBin. The routine continues looking until a detection is made or an echo has not been found after two hundred sweeps. When a detection is made the site profile is lifted using LiftProfile. If no echo is found a guess is made about where it might have been and a profile from there is lifted using LiftNoise. This starts at the same sweep as the Home site profile and the variable DetectSweep is set equal to IndexOne. A range bin three larger than the Home site is chosen to compensate for the link delays.

The phase comparison amplitudes are assembled into the Tin and Tos vectors with LiftPhase. This picks out the amplitudes from the same range bin and start sweep as the Home site profile. It directly lifts the values without any averaging.

This completes the lifting of data out of the AtoDMem. LiftMeteorObsv passes ObsvBuff back to the controlling program. The DMA hold flag that ensured the data was not overwritten while the observation was being lifted is then released. The save number and detection site is added and the record is transferred to the MetStore. The date of the event and any persistence of the ionisation trail is added to the record when the observations are written out to disk. Not reading the date at the time of detection saves time with very little added complexity. The duration of a persistent echo is not available at that time anyway.

3.4 Moving To and From MetStore

Moving observation records to and from the extended memory MetStore is done using the interrupt routine ExtMemMove. This uses the Move Block BIOS interrupt $15, subfunction $87. The transfer to storage above the 1 MByte boundary is done by the Intel 80286 Microprocessor operating in the protected mode. Interrupts are disabled during the operation so care must be taken to ensure that none of the start of sweep IRQ#3's are lost.

When an observation record is ready to be transferred to the MetStore the handling program tells the detector to make a transfer during the next convenient sweep. The boolean NeedToStore is set true. Provided the detector does not detect or release an echo in a sweep then there is enough time to make the transfer as well.

Transferring the 1280 bytes of ObsvBuff up to the MetStore takes approximately 1 ms. When DMA data transfers are running concurrently this time extends to approximately 1.15 ms. In the sweep an observation record is transferred to the MetStore the total time required by the interrupt detection routine detector is 1.75 ms. This is comfortably less than the time between each sweep.

These timing delays were measured by modifying the detection routine to always transfer to the store. By setting the I/O bit 0 on the interface board high before the transfer and low again after it was completed the elapsed time could easily be measured. Setting Port[DMACard] high to $F1 then low to $F0 and attaching an oscilloscope to line 1 of the DB25 plug on the timing board does this.

The Block Move BIOS interrupt $15, subfunction $87 requires the calling routine to set up a global descriptor table. A pascal definition of this Global Descriptor Table (GDT) is in the OrbLift unit. This contains the address locations for the protected mode operations. Most of it is preset to zero. Only the source and target Local Descriptor Tables need to be specifically set up using the SetGDTbones procedure.

Transfers are always made from or to the observation record ObsvBuff within the DOS accessible memory. Two GDT's are set up as part of the OrbLift initialisation section. One to provide transfers to the extended memory, GDTwrite, the other to read the records down from the MetStore, GDTread. The address locations of ObsvBuff are preset into these.

The actual transfers are done with the interrupt service ExtMemMove. This procedure is installed as interrupt $60 at the start of the observing program. When called, the boolean ToExt defines the direction of the transfer and which of the two GDT's to use. ExtMemMove makes a local copy of the Global Descriptor Table before adding the addresses and calling the BIOS Move Block. The MetStore address corresponding to the SaveNum location is used in writing to the store. ProcNum defines the record to be read down. If an error occurs in the transfer the carry flag is set. The register AH is used

to pass an error code. In this case the observing program is terminated with a Halt(3).

A more complete description of the entry and exit status of the registers can be found in the Technical Reference Manual for the IBM AT, Section 5-167 System BIOS.

The program `ExtMove.Pas` can be used to test the transfer of observation records to and from the extended memory. The program declares an equivalent 1280 byte BuffSize array on ObsvBuff using the Turbo Pascal absolute declaration. This makes it easier to inspect individual bytes within the ObsvBuff.

`ExtMove` first sets up the extended memory addresses in MetStore. It then fills each byte in the observation buffer with SaveNum. Each record from 1 to MaxStore is successively written up to the MetStore. The observation records are then read down and the contents checked. Any error detected causes the errant record to be dumped onto the screen.

3.5 Archiving Observation Records

At the end of each storage run the observation records in the MetStore are written out to disk. The procedure WriteOutStore in the OrbLift unit does this. The contents of the MetStore are read down into ObsvBuff using the ExtMemMove service routine on interrupt $60. Any record of echo persistence is added from the PersInfo array. At the start and end of each storage run the CMOS hardware clock is read to determine the date. The CMOSDate function in the GenUtil unit uses the BIOS interrupt $1A, subfunction $04 to do this. Comparing DateStart and DateEnd checks whether midnight has past during the storage run. The correct data is added to the record.

The observation records are then written to disk. Each day has a separate subdirectory in C:\ DATA. The observations from each hour go into a different file in that directory. The format for this is

C:\ DATA\ <yymmdd> \ NZST_<hh>

For example a meteor detected on 1989 May 16 at 0622 hours will be found in the file

C:\ DATA\ 890516 \ NZST_06

WriteOutStore requires interrupt channel IRQ#0 to be running to service the disk operations. It checks to see if the appropriate file already exists. If it does the file is opened and records are appended to it. Otherwise a new file and directory, if necessary, are created. To minimise opening and closing files the routine checks the hour of each successive observation record read from the MetStore. If it changes, the old file is closed and a new file opened.

After a period at Birdlings Flat the *luggable* MAC AT is returned to the Univserity so that observation files on disk can be transfered to the University

VAX mainframe for final archiving. The computer is connected to the ethernet and the data files transferred to disk. Originally this used DecnetDos but is now done by the file transfer protocol, ftp. The data uses a fixed record size of 1280 bytes. Once on the VAX network the data is written as a VAX/VMS backup saveset onto magtape. A DEC microVAX II/GPX acts a file server for the Physics Department 1600 bpi tape drive. A copy of the raw observation files NZST_**. are stored in the department. The screened observation record files, NZST_**.Orb are stored on a different tape. The reduced data records are stored on the 70 MByte hard disk of a second PC. This machine is a Cyclone 386-SX clone used as a program development, data reduction and graphics display machine. A backup of the reduced data is held on floppy disks.

Chapter 4

Radar Data Reduction

The radar system needs to be able to determine the position and velocity of a meteor at its time of observation. The azimuth, zenith angle and scalar velocity uniquely define the velocity vector of the meteor. The AMOR system consists of three spaced receiver stations. The transmitter, elevation interferometer and data logging computer all reside at the Home site. The meteor's position in space can be established relative to a coordinate frame centred on this Home site of the radar station.

The Birdlings Flat meteor radar has a narrow azimuthal antenna beam pointing south. The meteor's ionisation trail must be close to directly south or north to be observed. The range of the meteor can be determined by the time it takes the radar pulse to travel to and be reflected back from the ionisation trail . The elevation angle of the echo is determined by comparing the relative phase angle on two spaced receiving aerials at the Home site. The range and elevation fix the position of the ionisation trail formed by the meteor. Timelags are determined by comparing the echo profiles recorded from each of the three sites. Using the station geometry the timelags give a projection of the meteor velocity in the ground plane. A radar reflection only occurs when the meteor's ionisation trail is perpendicular to the reflected echo. The elevation angle fixes the plane the meteor was moving in. The meteor velocity can therefore be established.

4.1 Range

The distance from a receiving aerial to the meteor's ionisation trail is the range of a meteor. This can be calculated by measuring the time delay between the outgoing pulse and the reflected echo. The logic on the A/D Timer Control Board produces a series of 40μs range bin intervals. Every 66 of these pulses (2.64ms) the board sends a trigger pulse to the transmitter. Internal timing starts here. The corresponding A/D sample is labelled range bin zero (RB#0).

A series of time delays occur before the radio pulse finally leaves the trans-

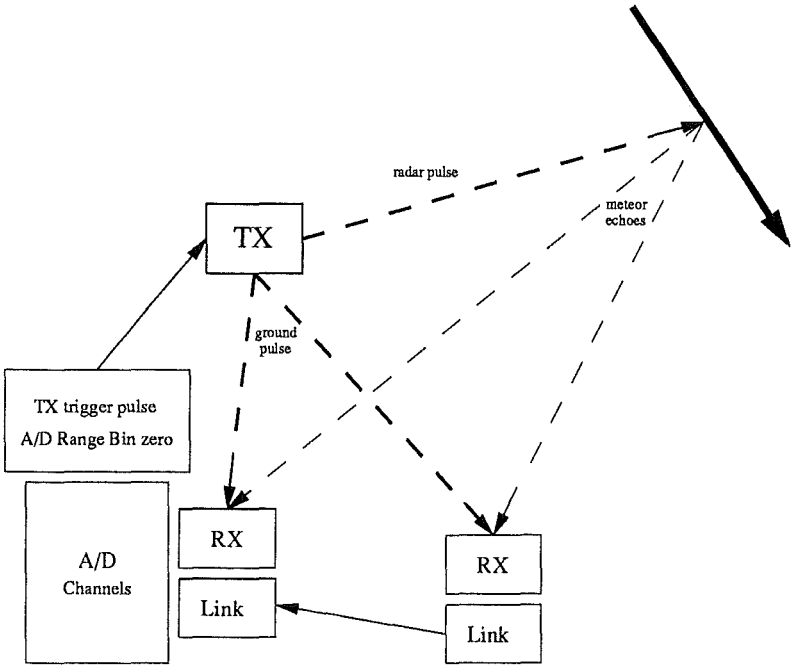


Figure 4.1: A schematic diagram of the time delays associated with measuring the range to a meteor trail. Both the path directly via the ground pulse and that of the reflection echo are shown.

mitting aerial. The trigger pulse must travel the 0.5 km down the transmission line to the transmitter hall. Various switching circuits in the transmitter (TX) produce a delay and the response time of the receivers (RX) before the A/D conversion records the data should also be considered. An FM data link sends the video signal from the two remote sites to the Home site to be recorded. This also needs to be included. Figure 4.1 gives a schematic diagram of these time delays.

Measuring the time interval between sending the transmitter trigger pulse and receiving the ground pulse directly from the transmitter will provide the best estimate of these delays. This time interval includes all the delays mentioned above in addition to the time it takes the ground pulse to travel from the transmitter to the receiving aerial. The system delays are tabulated in Table 4.1. The distances from the transmitter to the receiver sites can be estimated from Figure 4.2. The speed of light is assumed to be $c = 3.0 \times 10^5 \text{km.s}^{-1}$.

The radar pulse will travel 12 km in the 40μs interval. Each additional 40μs range bin therefore corresponds to a 6 km increase in the range. The range in kilometres can be expressed as equation (4.1). The range bin number

Figure 4.2: A topographic map of the station layout for the Birdlings Flat Meteor Radar.

Site	TX trigger to ground pulse (μs)	Distance TX to RX (km)	Ground pulse travel time (μs)	System Delay (μs)	Equivalent Range Delay (km)
Home	170	0.5	2	168	25
Nutt	300	10.5	35	265	40
Spit	280	8.2	27	253	38

Table 4.1: System timing considerations in calculating the range to a meteor trail. The timing delays were measured 1989 June 14. They are accurate to approximately 10μs. Note: the equivalent range delay is the two way *flight* time of the radar pulse.

is denoted RB.

$$\text{Range} = 6 \times \text{RB} - \text{Range Delays} \tag{4.1}$$

The measurement accuracy for these system timing delays is 10μs. It might be possible to refine this measurement using the range offset techniques described in the next section. This includes an allowance for the duration of the ground pulse. A variety of different transmitter power levels were used to make the measurement and delays measured to the ground pulse peaks were all included within this range interval of 10μs. The uncertainty is of the same order as the half power points for the 66μs radar pulse duration. The system has a 40μs resolution in the range bin timing. Prior to 1991 April, a measurement uncertainty should be allowed of 6 km in all range measurements. Since this date a rangebin offset has been calculated by the observing program. This is discussed in the following section. It should improve the range accuracy for the Home site range to near 1 km.

4.1.1 Range Scan Offsets

The rangebin recorded for a meteor echo corresponds to the bin with the greatest echo amplitude. The peak occurs somewhere within 20 μs of this point. The 66 μs echo pulse covers two or three adjacent rangebin samples. Echo amplitude information from the two neighbouring rangebins allows the echo peak to be placed more accurately. An offset of the recorded rangebin from the location of the peak is calculated for the Home site range scan[1]. The offset, calculated in microseconds is recorded in the ObsvBuff byte RangeOffset. This location in the observation record originally contained the DetectSite variable. Since Home site detection is used exclusively this variable was redundant. The

[1]This improvement to the range estimation followed as a natural extension of the investigation of the quarter, half, quarter averaging discussed in the next section. The code was added to the observing program on 1991 April 23. Yesterday as I write this !

RangeOffset value has 100 added to it to ensure that it is not confused with previous entries as DetectSite and to cater for negative offsets.

The shape of the receiver range scan is plotted in Figure 4.3. This shape is discussed more fully in Section 4.2.3. Every 40 μs an amplitude sample is taken. The echo peak can occur at any point with respect to these rangebin amplitude samples. The rangebin with the maximum amplitude could occur at any offset value from -20 μs to +20 μs. The amplitude associated with this was labelled, y. Taking moments about the maximum amplitude sample makes an estimate of the peak position possible. The two neighbouring rangebin amplitudes are labelled x and z for the earlier and later values respectively. By choosing appropriate constants with which to multiply each of the amplitudes, some account of the non-symmetric shape can be made. For a range offset, δ_r, and the peak amplitude, y_p, the relation was determined using equation (4.2).

$$ax + by + cz = y_p \delta_r \tag{4.2}$$

The coefficients need to be determined. For a series of offsets at 2 μs intervals from -20 μs to +20 μs the amplitude values and range offsets were tabulated from the measured range scan. A least squares fit was conducted to determine the best coefficients. The peak amplitude is not measured and should be converted in equation (4.2) to the recorded amplitude, y_r. From equation (4.6) these two are related, $y_p = 1.67 y_r$. The results are given below. The offset (μs) is the time displacement of the rangebin from the location of the peak amplitude.

$$\delta_r = \frac{1}{y_r}(15.5x + 1.3y - 18.2z) \tag{4.3}$$

Figure 4.3(a) shows an example of this fit. The location of the peak predicted by the offset fit is plotted as a spot. With no radio noise the peak is placed within 2 μs of the peak for all positions of the peak with respect to the maximum amplitude rangebin. Once the peak starts to saturate the receiver the offset method will break down. The relation of peak voltage to the recorded amplitude (equation 4.6) also changes. Figure 4.3(b) graphically demonstrates this inability to determine range offsets for a saturating echo. The observing code determines the range offset for a series of sweeps up the rising edge section until peak saturation is indicated at a recorded amplitude of 155 A/D units. It then stores a mean range offset for that meteor.

This rangebin offset allows a determination of the position of the echo pulse to better than 10 μs. Range determinations for the Home site should be accurate to 1 km. The ionisation trail represents an isolated radar target, so one kilometre range resolution should be possible despite the 66 μs pulse (10 km long) used. The range accuracy becomes limited by the ability to determine the timing delays associated with the system.

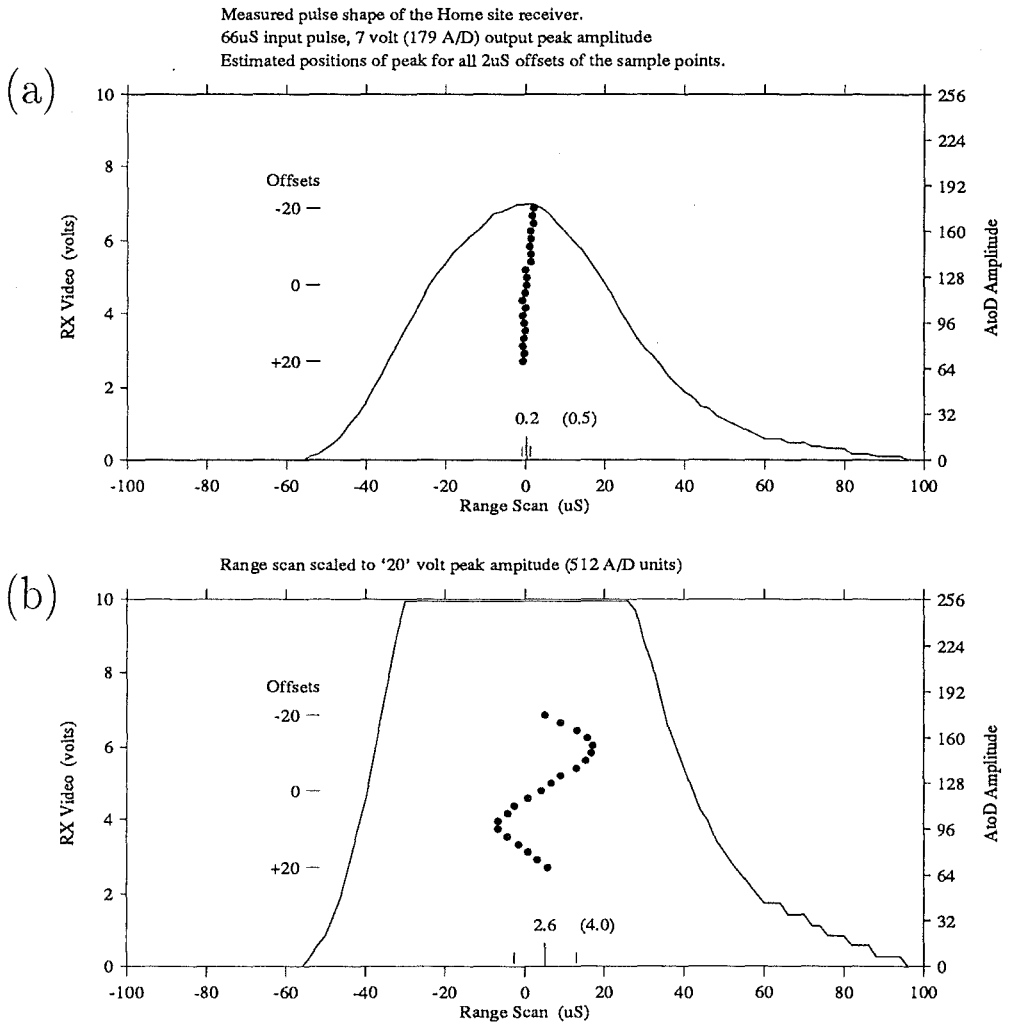


Figure 4.3: The three adjacent rangebin amplitudes are used to locate the peak amplitude more precisely. The predicted position of the peak is plotted for offsets of the maximum rangebin amplitude from -20 μs to +20 μs. The mean position (and standard deviation) of these estimates are noted at the bottom. (a) The range scan of a 66 μs pulse as measured. (b) Once the receiver saturates the estimate becomes unreliable.

4.2 Maximum Echo Amplitude

The maximum amplitude of a meteor echo gives an indication of the ionisation density in the trail. The antenna gain for echoes which come from the edge of the beam will be much less than those in the centre. No information is collected indicating the location of the echo in the beam so that only in a statistical sense can the echo amplitude be taken as a measure of the line density of the ionisation column. Clearly the echo strength depends on the transmitter power output. The density of ionisation produced by the meteor is a function of the mass of the particle, its speed and the zenith angle of the trail. By making assumptions about the particle density it is possible to estimate its physical size.

This section looks at the data reduction scheme used to calculate the maximum amplitude of any given echo profile. The individual points in the data profiles recorded by the AMOR system are the result of a quarter, half, quarter average across three adjacent range bins. The recorded amplitudes do not directly measure the peak echo strength. The maximum determined from the profile needs to be related to this peak echo strength. The receiver calibration information from Section 2.1.2 can then be used to give the strength of the reflected radio pulse at the receiver.

Many of the reduction routines in the AMOR package use profile amplitude levels as control checks. To ensure that these points are not markedly affected by radio noise a smoothed profile is used for this[2]. High strength noise impulses also need to be removed. These might come from car ignitions, pulsing of the 2.4 MHz Winds Transmitter or corona emissions from the local 11 kV power transformer. The receiver outputs are sometimes affected by noise spikes from corona. This is the most common source of noise spikes. Noise spikes can occur at any point in the profile. They should not be mistakenly interpreted as the maximum echo amplitude for either control purposes or estimates of the meteor size.

The noise spikes are removed and replaced with the mean value. These *despiked* profiles are used in all subsequent processing. A symmetrical triangular smoothing is done and the location and size of the maximum are found. The smoothed profiles are placed in an observation record, SmoothBuff. This is kept in addition to the raw (despiked) data and used mostly for control purposes.

The echo amplitudes of the remote site channels are subject to greater measurement uncertainties than for the Home site. The remote site echo amplitudes are transmitted through an FM link before being recorded by the AMOR system. The absolute amplitude recorded depends on the relative position of the FM base frequency. This drifts slightly with changes in external

[2]For example the echo decay time is calculated using points down to Noise plus five. A random point might drop below this level very early.

temperature around the transmitter[3]. Until December 1990 the baseline voltage level of the FM receiver was monitored and adjusted by the observer. During large meteor echoes the baseline of in particular, the Nutt site, tended to be *pulled up* because of inadequate D.C. restoration. See page 133 for a more complete description of this problem. This adds an additional uncertainty to the large amplitude (peak saturation) echoes recorded on the remote site channels. Since 1990 December the level has been clamped electronically reducing the amplitude uncertainties further. In preference the Home site maximum echo amplitude should be used to characterise the meteor's maximum echo strength.

4.2.1 Profile De-Spiking

Individual noise spikes typically last for a few tens of microseconds.[4]. They occur in random range bins and will not often occur in the same bin on consecutive sweeps. The profile despiking routine is designed to identify isolated echo amplitudes that are substantially different from the expected value. The routine DeSpikeProfile steps through and compares each sweep amplitude with that immediately preceding and following. Where the amplitude is more than 30 A/D units greater than the average of the two neighbours a noise spike is said to exist. For particularly noisy profiles this level is lifted to three times noise. The routine adjusts spike amplitudes to the average value.

Corona represents a very intermittent problem[5]. For long stretches it is absent entirely. It is usually associated with the output from the FM link receivers. On particularly bad days the Home site AM receivers are also affected. Figure 4.4 displays an echo profile with moderate corona. It illustrates the despiking routine. The raw echo amplitudes identified as noise spikes are ringed and the despiked values marked with crosses. The routine does a good job under most conditions. Some small amplitude spikes do remain.

Figure 4.5 gives an example of an echo profile more severely effected by corona. Where the noise spike occurs during a Fresnel pattern no attempt is made to fit the correction to the oscillations. In fact extreme Fresnel oscillations in some echo profiles are misidentified as noise spikes[6]. Two examples of consecutive sweeps with noise spikes can be seen in this profile. The routine is not designed to handle these. One is partially corrected. The relatively low

[3]See section 2.4, page 39 for extra details on the behaviour of the transmitter with environmental temperature changes.

[4]Some particularly bad corona would see groups of spikes several milliseconds long repeated at 50 Hz intervals.

[5]As you might expect none of the observations in my example files contained any evidence of corona! I now have a special test file DeSpike.Pen containing a set of corona plagued profiles.

[6]The period of the oscillations in all these cases is the same as the pulse rate of the radar. The sampling frequency is therefore likely to introduce aliasing anyway. This makes accurate identification of oscillations doubtful at this point.

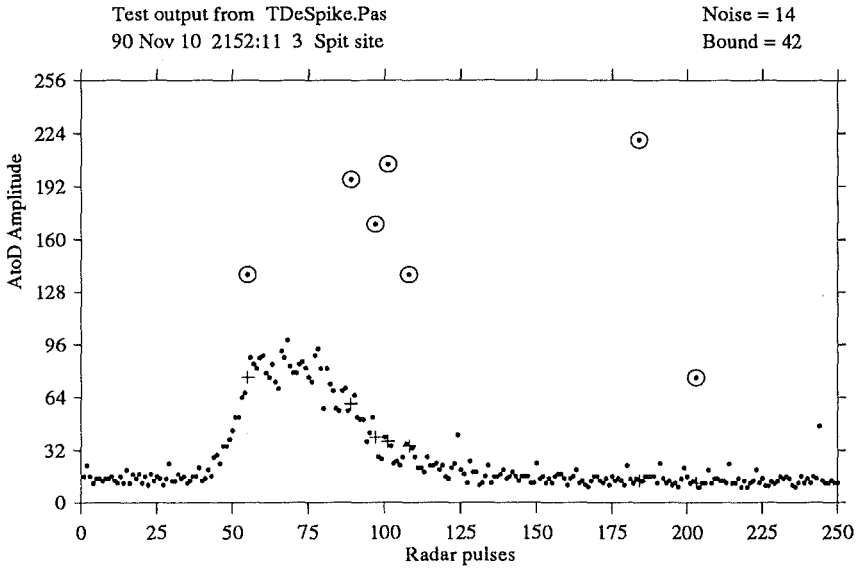


Figure 4.4: Diagnostic output from the DeSpikeProfile routine. The raw data are plotted as spots. Ringed points are identified as noise spikes by the routine and changed to the expected value. These are plotted as a small plus mark. These corona noise spikes are usually associated with the remote site channels. For this observation noise spikes are identified where the amplitude is 42 units greater than the average of the two adjacent neighbours. This being three times the noise level and greater than 30 A/D units.

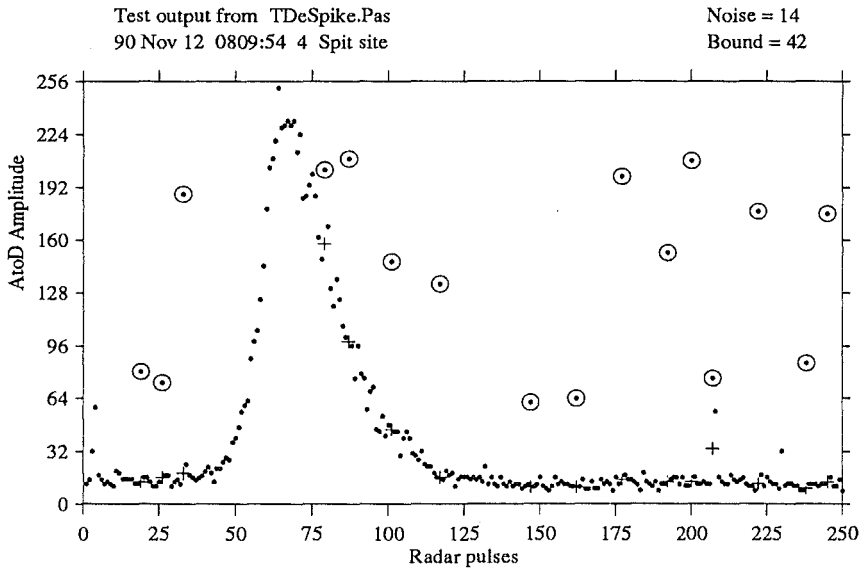


Figure 4.5: An example of the despiking for a profile severely effected by corona. The weak Fresnel pattern on the decay slope is compromised by the noise spikes. At index locations 3/4 and 207/208 corona noise occurs on two consecutive sweeps. The later example is partially corrected. The DeSpikeProfile routine was only designed to detect and fix noise spikes lasting for one sweep.

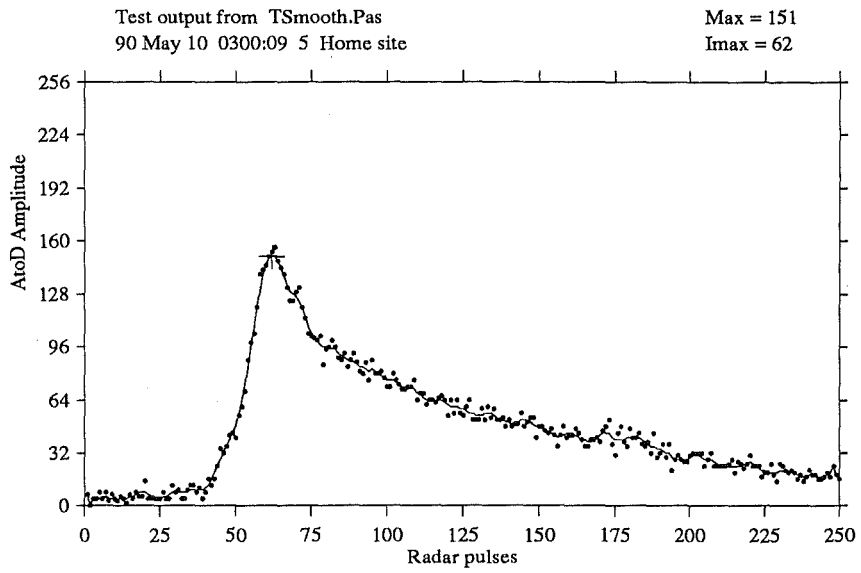


Figure 4.6: A triangularly weighted averaging is used to smooth the raw amplitude profiles. The raw data points are marked with spots and the smoothed profile overlaid. This smoothed profile is used for control purposes and to estimate the maximum echo amplitude. The maximum echo amplitude of 151 A/D units is marked. This occurs at sweep index 62.

amplitude of these ensure that neither affects the subsequent reduction.

4.2.2 Triangular Smoothing

A Pascal triangular smoothing algorithm is used to smooth the raw amplitude data. Each point is smoothed by considering the three points on either side. The averaging is weighted to give the nearer neighbours greater significance. Mathematically for a raw echo profile, y, the smoothed amplitude at each point is given by s_i in equation (4.4). For the smoothing range chosen ($R = 3$) this reduces to equation (4.5).

$$s_i = \frac{1}{(R+1)} \sum_{j=-R}^{R} \frac{R - |j| + 1}{R+1} y_{i+j} \tag{4.4}$$

$$s_i = \frac{1}{16} (y_{i-3} + 2y_{i-2} + 3y_{i-1} + 4y_i + 3y_{i+1} + 2y_{i+2} + y_{i+3}) \tag{4.5}$$

Figure 4.6 shows an example of the TriSmooth smoothing on a Home site profile. Effectively the $R = 3$ range of the smoothing function means no more than three consecutive points are above or below the curve at any time. At the ends of the profile the range of the smoothing is successively reduced to

ensure that the bounds, y_1 and y_{250} are not exceeded. The routine does a good job of smoothing the noise[7]. Calls to the TriSmooth routine specify the smoothing range to be used.

4.2.3 Radio Frequency Echo Voltage

The maximum echo amplitude is determined by the FindMax routine using the smoothed curve. The profile maximum will tend to be slightly underestimated especially for echo profiles with sharp rises followed by sharp decay sections. Using the raw profile directly would however be far too sensitive to interference from noise fluctuations. This maximum is the largest echo amplitude recorded. It is not the peak echo strength for that sweep.

To infer the peak echo strength a picture of the range scan of a radar pulse is needed. The AMOR system uses a 66 μs radar pulse. The 20 kHz receiver bandwidth ensures that the shape of this pulse shape is determined by the receivers. Figure 4.7(a) shows the amplitude scan of a 66 μs pulse feed into the Home site receiver. The peak amplitude was set at 7 volts video output.

A returning radar pulse can fall in any position relative to the 40 μs sampling points. Ideally the recorded echo amplitude should not be too sensitive to variation in this position. A quarter, half, quarter average centred on the rangebin with the largest amplitude is taken. The recorded amplitude for that sweep will be less than the peak echo amplitude. Figure 4.7(a) shows an amplitude scan of a reflected radar pulse. The curve running from -20 μs to 20 μs shows the A/D amplitude that would be recorded given the offsets with respect to the peak amplitude. As can be seen the recorded amplitude depends only slightly on the position of the peak in the rangebin. The peak amplitude is 7 volts video (180 A/D units) while the amplitude recorded for this sweep would be 107 A/D units.

The radio receivers begin to saturate where the echo strength rises above about 35 μV[8]. Until this level the recorded amplitudes are linearly related to the peak amplitude. As the echo strength continues to rise the amplitudes in the adjacent rangebins also rise whilst the central value remains constant at 255 A/D units. The relation becomes non-linear. Even though the receiver is saturating the peak amplitude can still be inferred. Figure 4.7 shows this transition to a saturated echo.

The recorded amplitude as a function of the peak echo amplitude has been calculated. The relation is plotted in Figure 4.8. The curve remains linear until the echo peak begins to saturate at about 155 A/D units. For a recorded amplitude y_r, the peak echo amplitude y_p, in this section can be expressed as

$$y_p = 1.67\, y_r \tag{4.6}$$

[7]The steeper slope of the rising edge appears to be less affected by noise. The mean distance of data points from the smoothed curve is less in this region. The station timelags of the AMOR system are based on this section of the profile.

[8]The receivers themselves remain linear from noise (2 μV) up to this level.

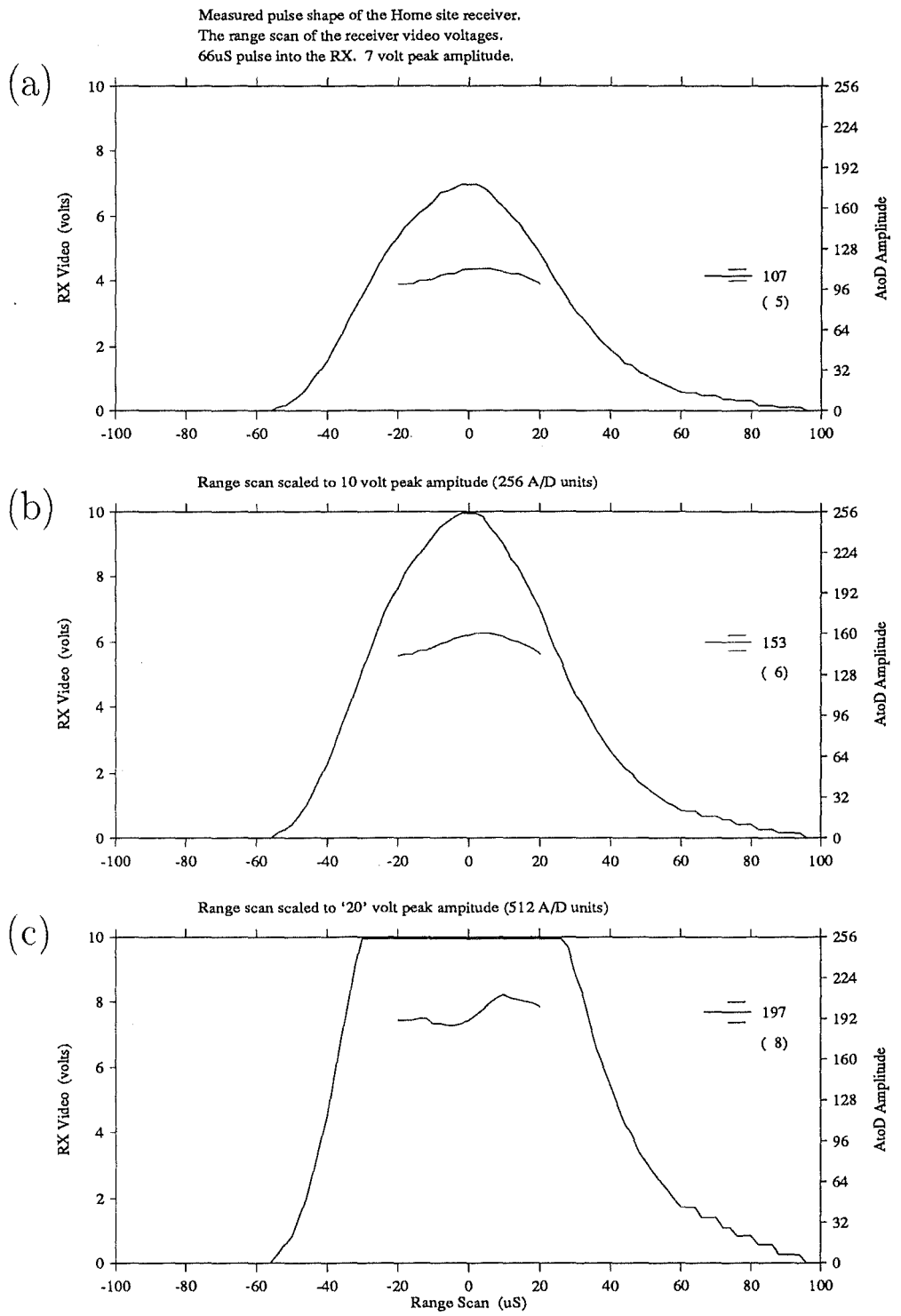


Figure 4.7: The Home site amplitude range scan for a variety of peak echo amplitudes. A/D rangebin samples are taken every 40 μs. These can be offset by any amount from the peak. The results of the quarter, half, quarter average used for the recorded amplitude are plotted for a series of such offsets. (a) The range scan of a 66 μs pulse as measured. (b) Echo strength corresponding to the onset of peak saturation. Increases in the recorded amplitude are non-linear from this point. (c) The signal strength is doubled to 512 A/D units. The recorded amplitude continues to rise with the contributions from neighbouring rangebins.

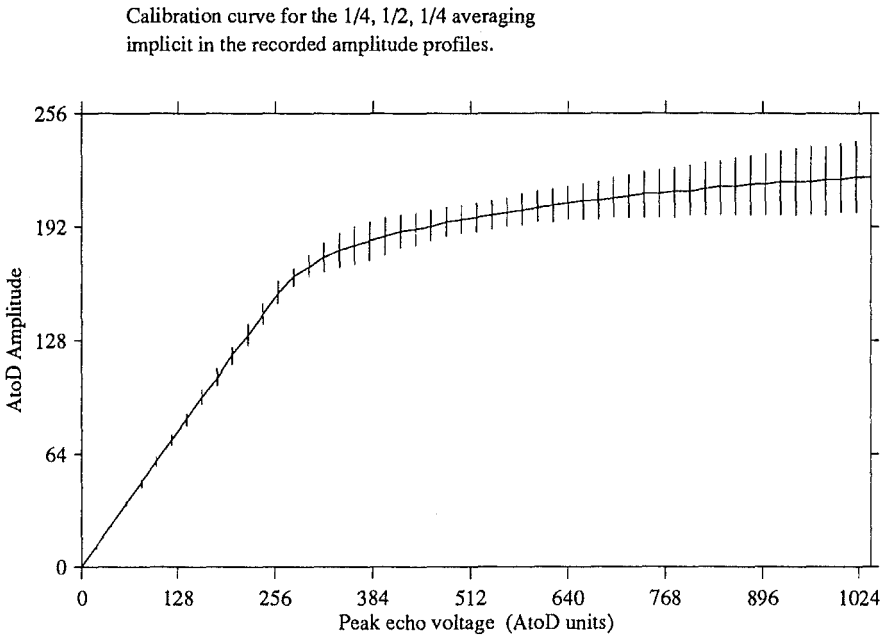


Figure 4.8: The amplitude recorded by the AMOR observing system is found using a quarter, half, quarter average across the echo range scan. This recorded amplitude is related to the peak echo amplitude in this graph. The relation remains linear until the peak amplitude begins to saturate the receiver. This corresponds to a recorded amplitude of about 155 A/D units. The error bars reflect the unknown location of the peak within the rangebin with maximum amplitude.

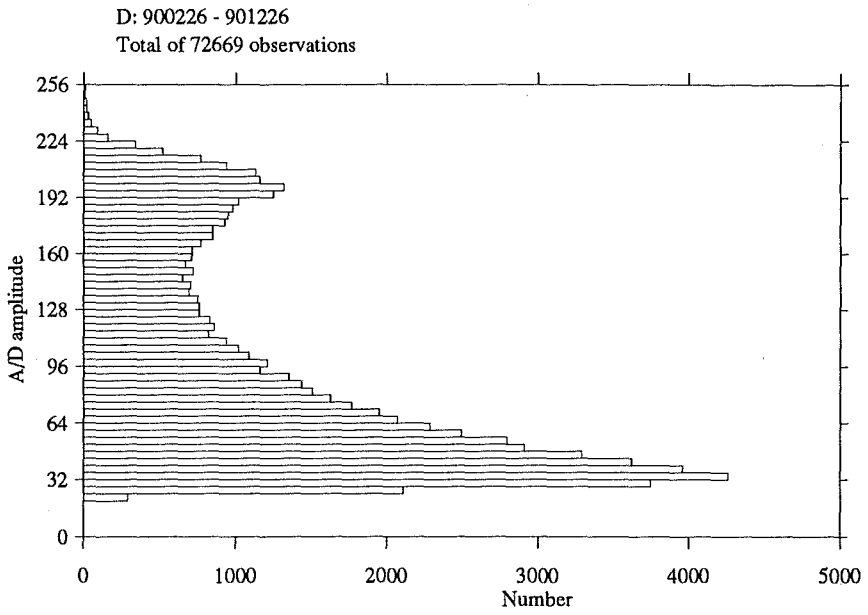


Figure 4.9: Cumulative distribution of the maximum amplitudes recorded at the Home site. The number decreases logarithmically from the detection level of 30 A/D units to about 155 A/D units. This later point corresponds to peak saturation. Above this the relation of recorded to peak amplitude becomes decidedly nonlinear, see Figure 4.8.

As the echo strength continues to rise the uncertainty about where in the rangebin the peak occurs becomes more pronounced. The uncertainty in the conversion from recorded amplitude to peak amplitude will correspondingly increase. The peak amplitude can still be inferred by the amplitude of the adjacent rangebins. Figure 4.8 provides a calibration curve that can be used to convert the maximum amplitudes calculated within the reduction package to peak echo amplitudes. The receiver calibration curves from Section 2.1.2 will convert these to R.F. echo strengths in μV at the receiver input.

Figure 4.9 plots the cumulative distribution of recorded Home site maximum amplitudes. The number of meteors with a given amplitude drops logarithmically until about 155 A/D units. This is exactly where the peak echo strength starts to saturate the receiver. The relation of echo strength to recorded maximum amplitude becomes non-linear as individual echoes saturate the receiver to a greater or lesser extent. In a statistical sense the the RF voltage at the receiver for these saturating echoes can still be determined.

The recorded amplitudes of the three receiver channels are effected by peak saturation in a different manner. Essentially the recorded amplitude depends

on the adjacent rangebins. The shape of the range scan in the wings of the pulse are critical to this. The wings are very sensitive to the bandwidth of the receivers and this is slightly different for each channel. The baseline D.C. restoration problem (see page 80) of the remote sites complicates the issue. The peak of these saturating echoes occurs at amplitude levels of about 200, 255 and 230 A/D units for the Home, Nutt and Spit channels respectively.

4.3 Elevation Angle

The elevation angle along with range information enables the position of the meteor trail to be determined. Since the ionisation trail must be nearly perpendicular to the geometric path of the radar pulse the elevation angle also fixes the plane the meteor was moving in.

Two receiving antenna arrays at the Home site are used as an interferometer to measure this elevation angle. The two signals are fed into separate phase linked receivers. The outputs from these are compared to determine the relative phase angle of the echo at the two aerials. The comparison hardware gives a linearised sine (Tin) and cosine (Tos) output from which the relative phase angle can be calculated.

The two Home site aerials are spaced 5 wavelengths apart.[9] Beginning with an echo from the local zenith as zero degrees, the phase angle goes through four complete cycles down to an echo returning along the ground plane. These phase angles correspond to a change in the elevation angle from 90° to 0°. Echoes returning from the north vary from 90° to 180° in elevation. The phase angle is ambiguous in the sense that it cannot uniquely identify the correct elevation angle. The uncertainty about whether the meteor trail is to the north or south of the Home site also needs to be resolved.

The meteors detected by this radar system ionise in the altitude interval from 70 to 120 km. Meteors which are closer in range will be detected at higher elevation angles. Given the range to a meteor trail and requiring it to be in the given altitude interval allows the phase angle ambiguity to be resolved. The geometrical constraint of detecting ionisation trails only at right angles to the direction of motion means that those meteors moving from north to south across the ground are detected from towards the south. Meteor profiles that appear on the Nutt site before the Spit site are therefore characteristic of trails to the south.

The signals received at the two Home site antennas travel through different lengths of transmission line to their respective receivers. This adds an additional differential phase shift. To allow for this and any inherent phase variation in the receivers, a phase calibration constant is added to the relative phase angle. The radar frequency, transmitter peak power and aerial configu-

[9] In the reduction code I use a figure of 4.96λ for the separation. This extra accuracy is not really justified. The difference represents a distance of half a metre in 60 m.

ration are expected to give an altitude distribution peaking at around 97 km. A phase calibration constant is chosen to do this.

Meteor trails can be detected from ranges greater than 400 km.[10] The echoes from these trails arrive back on the sweep after the transmitter pulse was sent. Much of this range ambiguity can be resolved by the strategy outlined above. Some ambiguity remains. Meteor trails at 9°, 560 km cannot be distinguished from 38°, 150 km echoes. About half the observations in this ambiguous group come from each source. Assuming that all meteors are detected in the closer category renders observed velocities of above 80 km.s^{-1}for even the slowest meteors that actually come from 9° elevation. All meteors with phase angles in the range 310° to 350° and an observed velocity above 80km.s^{-1}are therefore rejected for further processing.

Meteors at these larger ranges have elevation angles of less than 13°. This increases the likelihood that the meteors were moving in a track very steeply inclined to the ground plane. The radar measures the component of the meteors velocity in the ground plane. For these low elevation cases the system is restricted to measuring only a small component of the total velocity. This substantially reduces the accuracy of the observed meteor velocity. Observations of meteor trails from beyond 400 km therefore are rejected.

4.3.1 Relative Phase Angle

The signals detected by the two Home site receivers are compared to give a relative phase angle. The two video outputs are combined giving a linearised sine and cosine expression of the phase angle. The phase outputs are called Tin and Tos respectively. The hardware do do this is described in Section 2.1.2, page 20. Where the signal is just radio noise the phase angle at the receivers will be random. Under these conditions the Tin and Tos outputs just give a random scatter of values. When a radar echo arrives the signal will have a consistent corresponding phase angle. The Tin and Tos values in the corresponding range bin lock onto their values associated with this phase angle for the duration of the echo.

To improve the accuracy of the phase determination the data from several pulses is used to give a mean Tin and Tos value. While the echo amplitude remains above 16 A/D units the averaging routine continues including Tin and Tos values from successive radar sweeps. Atmospheric wind shears will move the specular reflection point up or down the trail. Where this occurs the phase angle of the received echo will also move accordingly. This is a relatively slow change in the phase angle. Hence to avoid contamination from atmospheric effects only the first 24 'useful' data points are included.

Some echo power will be scattered back to the receiver before the meteor

[10]In fact no echoes are detected in the range interval between about 350 and 460 km. This gap corresponds to the sending of the next transmitter pulse and reseting the computer for the next range sweep.

has reached its specular reflection point. As the meteor moves toward this point of closest approach the direction of the echo pulse will also move. The phase angle will therefore change fairly rapidly as the meteor moves up to its specular reflection point. To avoid this effect the averaging of the Tin and Tos vectors is not started until the echo amplitude has risen above 70% of the maximum. This 'hook' in the phase angle versus time is only present for larger amplitude echoes.[11]

As an example of the Tin and Tos reduction process a reasonably large amplitude meteor observation is displayed in Figure 4.10. In the random noise at the start of the profile the Tin and Tos values exhibit a fairly random scatter. As the echo profile rises out of the noise these values lock onto the signal. The Tin and Tos outputs change fairly rapidly as the amplitude moves up the rising edge of the profile. During the decay section of the profile the mean Tin and Tos values drift slowly in time. The amplitude variations in the decay slope also suggest the trail is being affected by wind shear. See the MSc thesis *Atmospheric parameters from three-station radio-meteor data* for a more complete discussion (Howick, 1991). The actual data points used to calculate the means are marked with larger spots.

The actual variation of the Tin and Tos functions with phase angle has been measured, see Figure 4.11. To do this the two receivers were attached to aerials at opposite ends of a long plank. The plank was mounted so that it could freely rotate. A small 26.36 MHz transmitter was placed on Bayleys Road, about a kilometre from this aerial assembly. By rotating the plank the difference in path length from the transmitter to each of the receiver aerials could be varied. The corresponding voltage outputs on the Tin and Tos channels were measured. These voltages range from 0 to 10 volts and correspond to an 8-bit A/D amplitude value ranging from 0 to 255. The output voltages did not repeat every 11.38m,[12] but rather over 11.15m. The velocity of the ground wave at this frequency is about 2% below the speed of light. The origin of this difference has not been found; it remains a mystery. The variation of the phase angle over one cycle was fitted to the shorter wavelength. This measurement was done in November 1983 (Steel, unpublished). Since this date the radar frequency has been changed to 26.2 MHz. The calibration data is contained in the text file Phas8311.Cal.

The phase angle of the echo is calculated by performing an empirical fit on the mean Tin and Tos values. This is accomplished by the routine Fit-Phase_1983. The general scheme is to fit the angle to one of the mean values and use the other to determine the quadrant. If the average Tin value falls

[11]This shift in phase angle as the meteor approaches its specular reflection point could be used to estimate the velocity of the particle. It would provide another check on the velocity determination scheme used by AMOR.

[12]The radar was operating on 26.36 MHz. Channel zero of the citizen band was moved onto this frequency in 1989. The move was discovered in an amateur radio magazine. After inquiries to the New Zealand Radio Frequency Service we were reallocated 26.2 MHz.

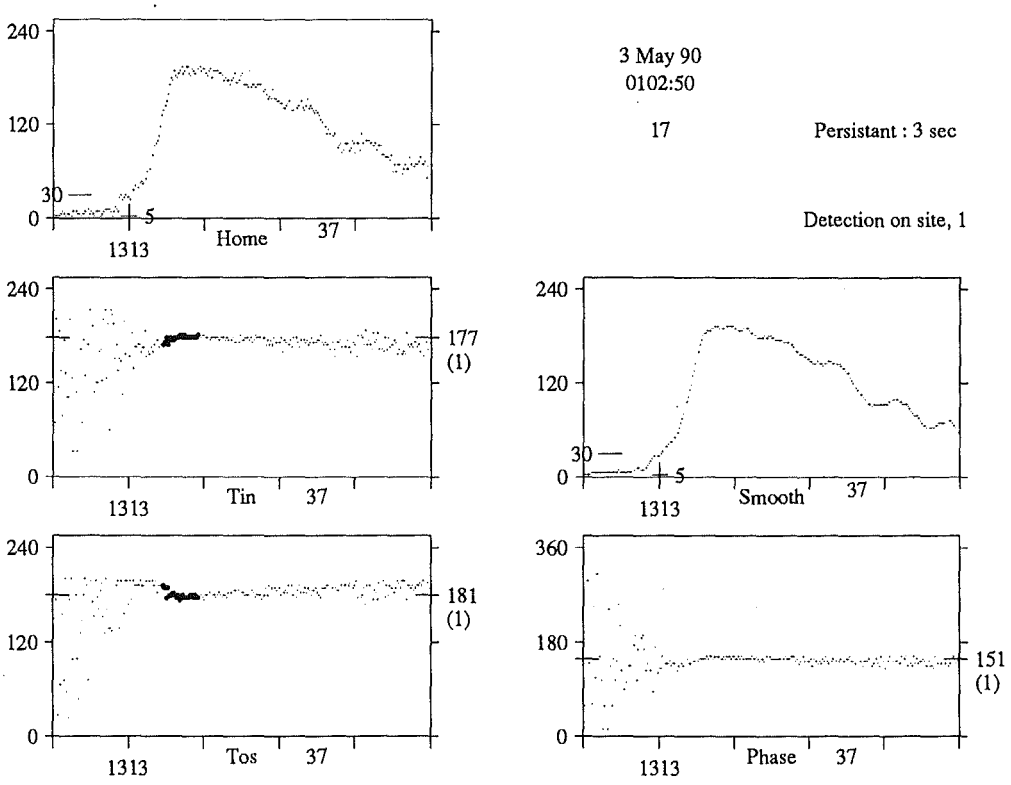


Figure 4.10: Test output for the phase calculation routines. The Home site amplitude profile, Tin and Tos outputs are plotted. The points used to calculate the mean Tin and Tos values are marked with larger spots. The smoothed amplitude profile is used to reduce noise effects. The phase angle as calculated for individual sweeps is plotted bottom right. The phase angle and the mean Tin and Tos values are marked, with error estimates in brackets. The display is produced by the program `TGetPhas.Pas`.

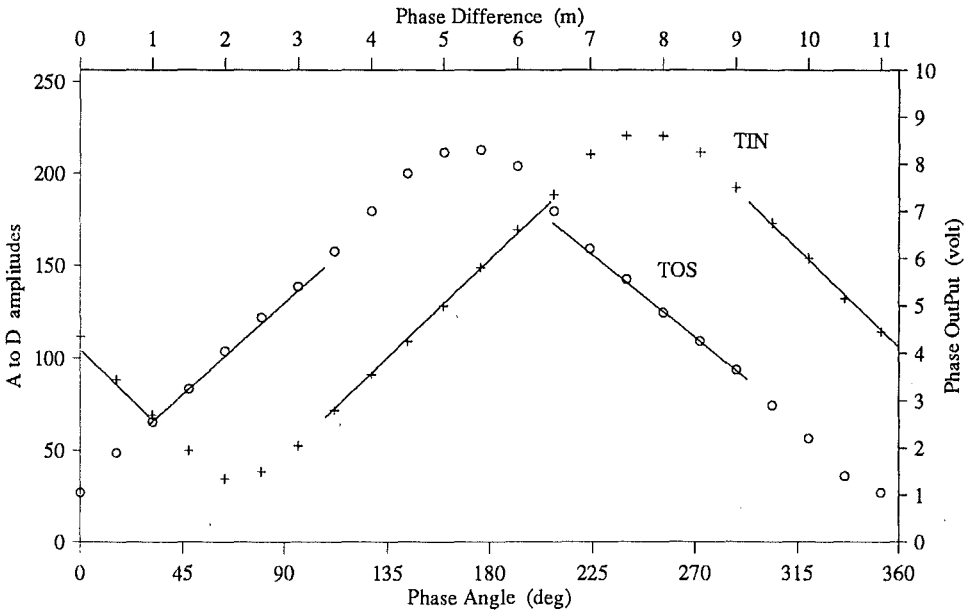


Figure 4.11: The Tin and Tos outputs plotted as a function of the phase angle. The interpolating lines are the linear fits used to calculate the phase angle. The second output at that phase angle is used to evaluate the quadrant. The phase difference was measured as the separation of the aerials. The 11.15m wavelength of the ground pulse provides one complete cycle of phase angle. The calibration data plotted here is held in the file Phas8311.Dat.

in the linear section of its variation then it is used to calculate the angle and the average Tos value resolves the quadrant. For values outside that range the roles are reversed. In terms of Figure 4.11 the interpolating lines are used to give the angle whilst the other phase output gives the quadrant. The 'Phase' plot on Figure 4.10 shows the phase angle at each sweep as determined by the individual Tin and Tos values. It provides a graphical display of stability in the phase angle determination.

On my first implementation of this empirical phase fitting routine I put the linear interpolations of the Tin and Tos functions as nearly as possible through the measured data points. This left holes in the resultant phase angle distributions. Figure 4.12 shows a large gap in phase angles around 210°. Another can be discerned at 105°. These gaps align closely with the transitions from fitting the angle on one phase output to fitting on the other. To try and render this empirical phase angle determination more nearly seamless I adjusted the linear interpolation lines slightly. The final result is Figure 4.13. The gaps have disappeared without significantly compromising the original fit of the raw calibration data.

The gaps evident in Figure 4.12 were around 10° wide. This gives a limit to the accuracy possible in determining the phase angle of a meteor echo. The standard error of the mean is calculated for the mean Tin and Tos values. An uncertainty for the phase angle due to the variation in the average Tin and Tos values is estimated within the Get_Phase routine. This uncertainty corresponds to a shift of one standard error in the mean value used to fit the phase angle. If this phase difference is greater than 5° then the observation is considered to have too large an uncertainty in the phase angle determination and is rejected. This provides a two sigma confidence level, that any error from determining the average Tin and Tos values, is less than the 10° uncertainty in phase angle implied above.

4.3.2 Phase Range Distribution

Given the relative phase angle between the Home site receivers it is possible to calculate the elevation angle. A schematic diagram of how this is done is shown as Figure 4.14. An echo returns from an elevation angle Ψ. The path difference travelled to reach the two aerials is an integral number of wavelengths and an additional fraction of a wavelength. This additional fraction produces the relative phase angle of the signal at the aerials.

Given a relative phase angle, ϕ measured at the aerial, the path difference, p.d. is

$$\text{p.d.} \quad = \quad n\lambda + \frac{\phi}{2\pi}\lambda \tag{4.7}$$

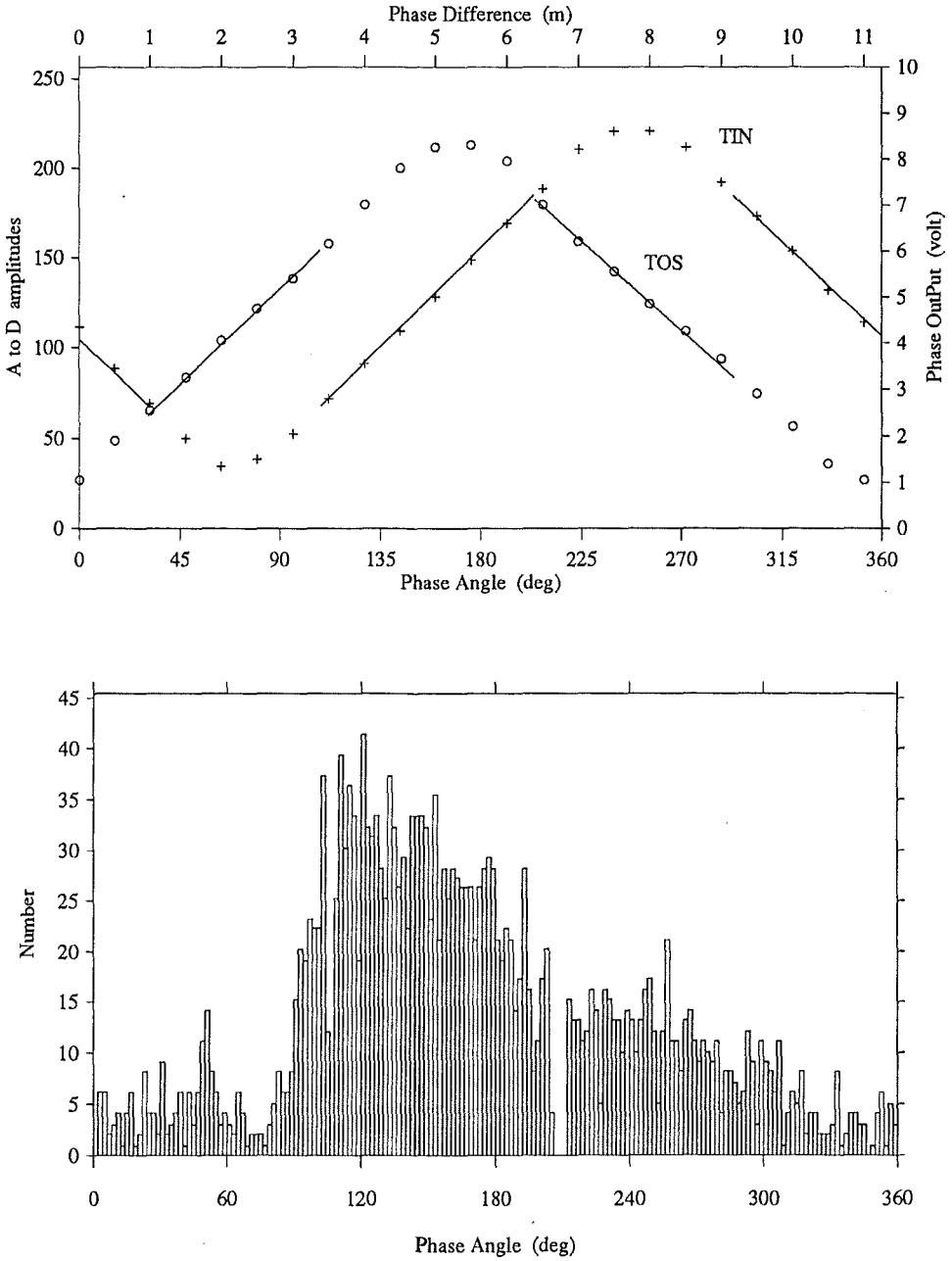


Figure 4.12: The phase angle distributions for two similar empirical fits to the Tin and Tos calibration data are compared in this diagram and in Figure 4.13. The original fit placing the linear interpolation through as many of the calibration points as possible is plotted here. This left holes in the phase angle distribution at 105° and 210°. These are associated with the transition between fits. The phase angle distribution includes 2244 observations from 1990 June 4.

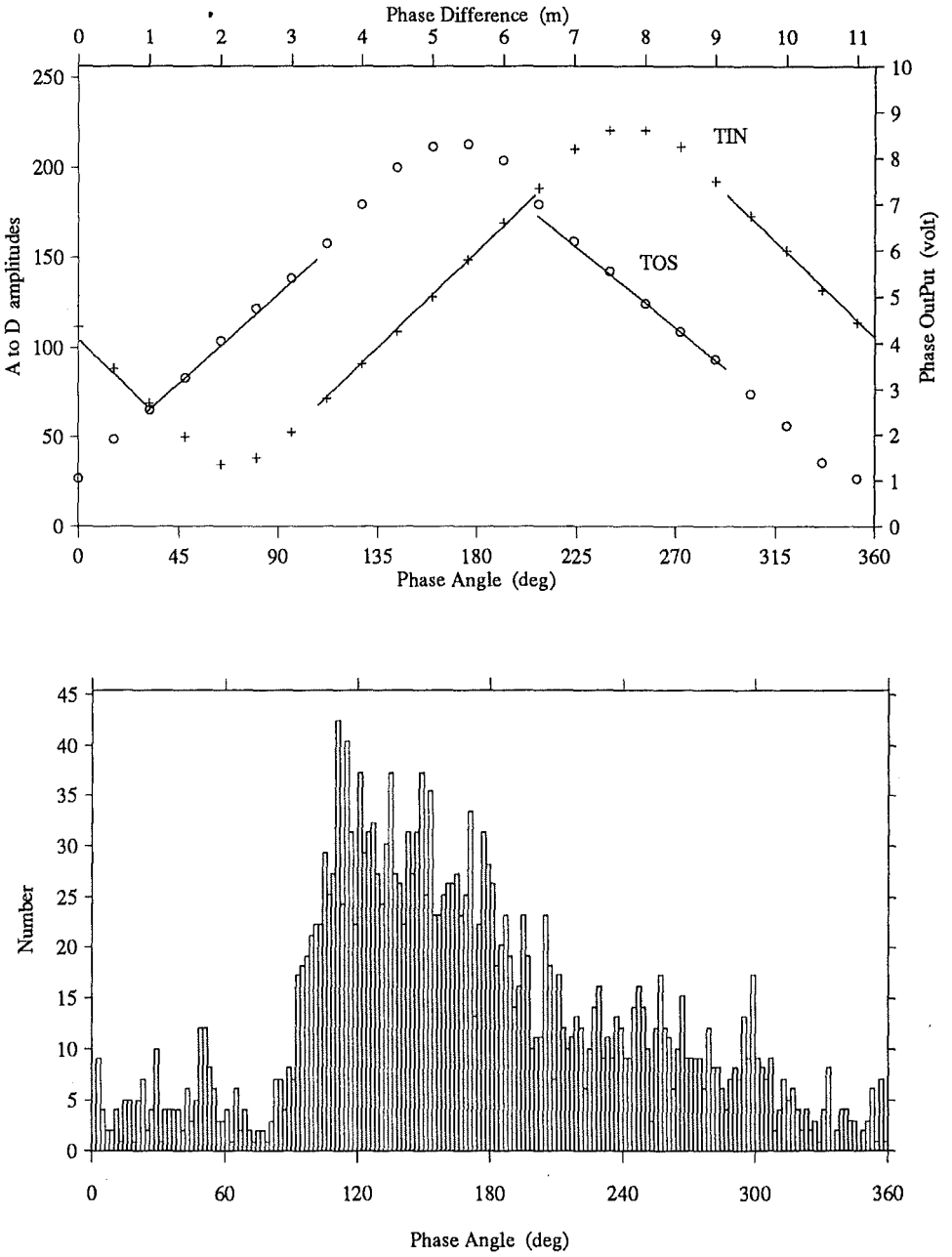


Figure 4.13: The phase angle distributions for the current empirical fits to the Tin and Tos calibration data. By making slight adjustments to the fit from Figure 4.12 the final distribution can be rendered more nearly seamless. The phase angle distribution includes 2244 observations from 1990 June 4.

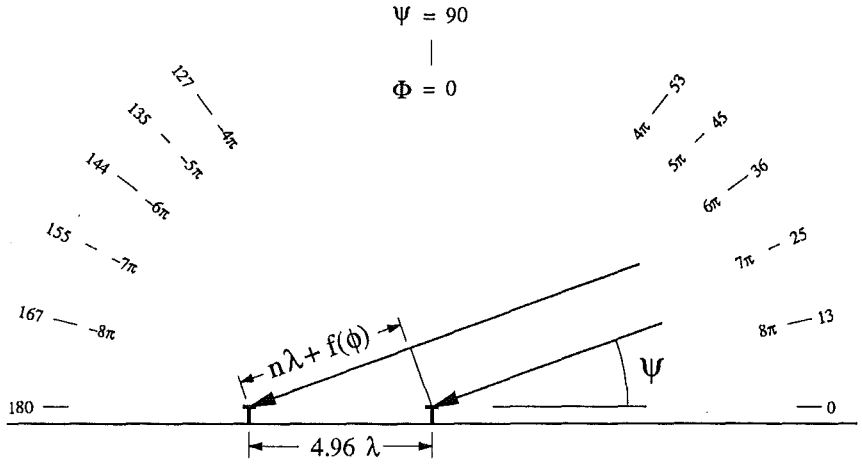


Figure 4.14: Sketch of the relationship between the echo elevation and the relative phase angle. Zero phase difference and 90° elevation is defined as an echo from directly overhead. The scale shows total phase angles in radians and corresponding elevation angles in degrees. The central portion is an expanded schematic of the Home site. An echo returning from elevation angle Ψ gives a path difference $n\lambda + f(\phi)$.

The total phase angle, Φ of the echo can be thought of as $2\pi n + \phi$. The elevation angle, Ψ is

$$\begin{aligned} \cos\Psi &= \frac{\text{p.d.}}{4.96\lambda} \\ &= \frac{n + \frac{\phi}{2\pi}}{4.96} \end{aligned} \tag{4.8}$$

A number of issues complicate the problem of determining the elevation angle. There are different lengths of transmission line connecting the two receivers to the aerial arrays. The relative phase measured at the receivers needs a phase calibration constant added to become the relative phase angle at the aerials. Since the aerials are spaced 5λ apart, n can take any integral value from -4 to 4. The question of which particular phase cycle and which elevation quadrant the echo is in, needs to be addressed. The distribution of values of echo range against the relative phase angle of the echo (Figure 4.15) gives a platform from which these issues can be considered. The phase angle values on the phase range diagrams in this section have all had an appropriate calibration constant added.

The data points in Figure 4.15(a) are plotted directly on the phase range diagram[13]. For a set of reflection points at 100 km altitude the corresponding

[13]My very own 'H-R Diagram' to prove to the astronomers that I am one of THEM!

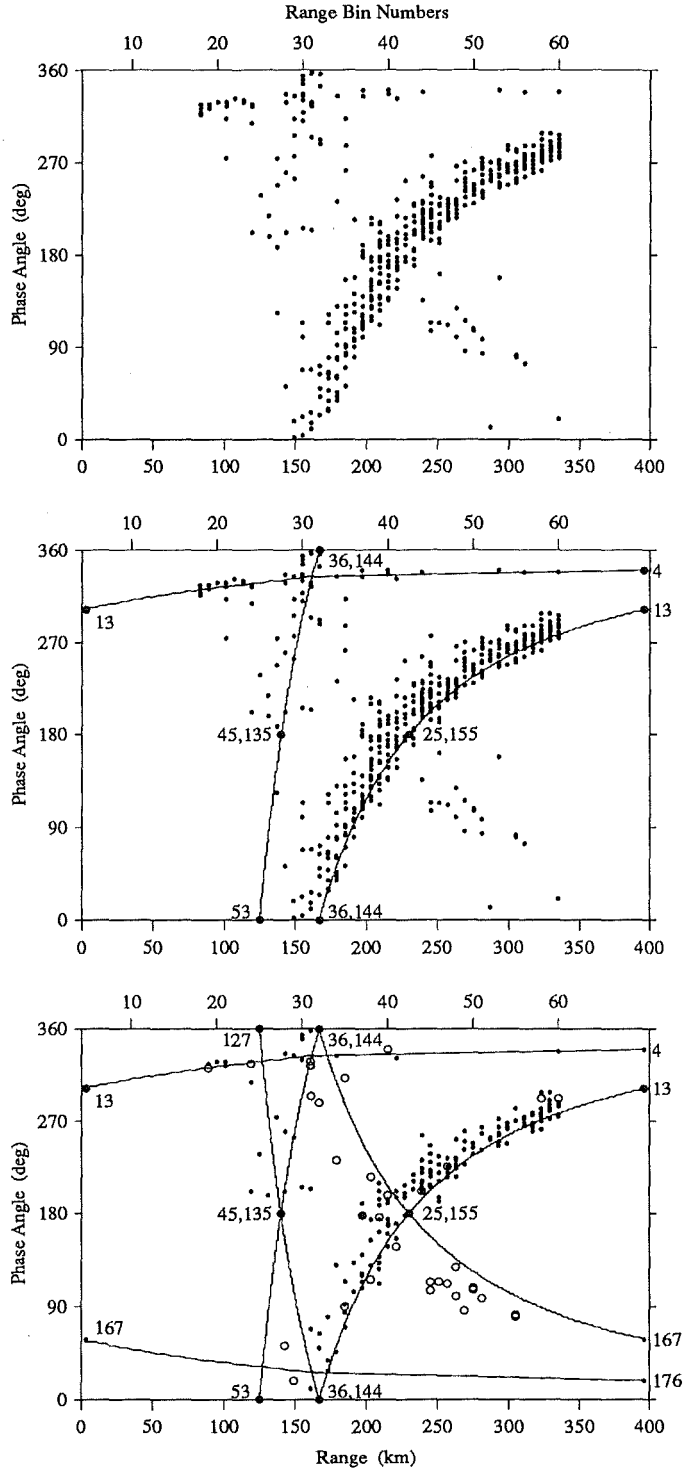


Figure 4.15: The phase range distribution with 100km constant altitude curves superimposed. The test data used for these plots are contained in the observation files D:\ 900604 \ NZST_03.Orb and NZST_07.Orb. (a) The raw data plotted directly on the graph using the program RawPhasR.Pas. (b) 100 km altitude contours for echoes from ionisation trails to the south. (c) Echoes identified as from the north are plotted as open circles. The altitude contours are extended to northern reflection points.

relative phase angle and range to the trail can be calculated. This was done incorporating a curved Earth geometry. A set of 100 km altitude curves for points to the south of the radar station are overlaid in Figure 4.15(b). The implied elevation angles at various points are marked on the curves. The constant altitude curves follow the observed distribution. Most of the meteor trails detected by this radar would reach maximum ionisation slightly lower than this 100 km altitude level. For a given phase angle, hence elevation, lower reflection points will occur at shorter ranges. This is precisely what is observed. Radar echoes from meteor trails at ranges greater than 400 km will appear on a subsequent sweep. The curve from elevation angle 13° to 4° fits these points. Ensuring that these points are at approximately the expected altitude provides the most sensitive fit of the phase calibration constant. A number of points do not follow this picture. Continuing the 100 km altitude curves for echo points to the north fit the remaining points, Figure 4.15(c).

By using the information inherent in the phase range distribution most of the ambiguities mentioned above can be eliminated. Observations that fall near crossing points of the constant altitude curves continue to remain in doubt. The reflection geometry requires that meteor trails to the south of the station were produced by meteors travelling from north to south. The two remote sites are aligned approximately north-south[14]. Echo profiles for meteors moving south will occur on the Nutt site first. Meteor trails to the north will occur on the Spit site first; these points are plotted as open circles on Figure 4.15(c).

Most of the radar echoes occur between elevation angles of 13° and 36°. As a first guess the trail is assumed to lie on this altitude curve and the elevation angle calculated accordingly from Get_Elevation. For phase angles greater than 320°, n = 3 is assumed rather than 4. Get_Altitude gives the altitude for this elevation angle. The information is passed to the routine ResolvePhase to check the results. If the time lag (Lag23) between the remote sites implies a north echo the values are recalculated. The implied meteor altitude is compared with the altitude range 70 to 120 km. Where the result falls outside this range the elevation angle is recalculated assuming a new value of n. Where the altitude still falls outside the expected range the observation is rejected as having an unresolved ambiguous phase angle.

Figure 4.16 details the results of this process. Each of the symbol types represent a different set of assumptions about the phase and range ambiguities. Echoes from trails at short ranges can fall within the altitude range and still not provide a unique elevation angle. All observations that fall in this region, $n < 3$, are rejected.

A problem remains in trying to distinguish 9° elevations where the echo

[14]It is only by the strangest quirk of circumstance that the two remote site stations are aligned north-south. Both stations have been rebuilt several times and moved at least twice. Station 2 originally dubbed the *Swamp* site was placed directly north of the Home site when first built.

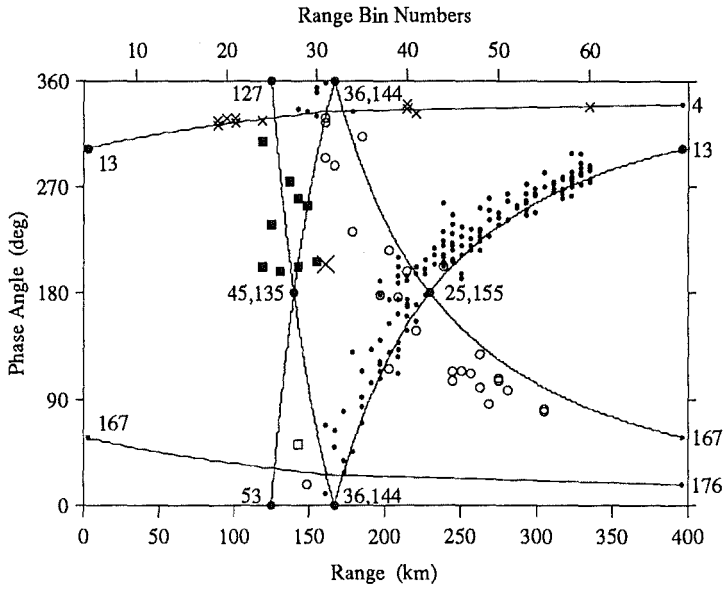


Figure 4.16: The phase range diagram with different resolutions of phase angle itemised. The various identifications of the symbols are in the table below. The large plus and cross symbols remain ambiguous. This graph is the diagnostic output for the routine ResolvePhase. It is produced by the program TResolve.Pas.

South (Lag23 $\leq$ 0)		North (Lag23 > 0)	
•	$13 < \Psi < 36$	o	$144 < \Psi < 167$
■	$36 < \Psi < 53$, n = 3	□	$127 < \Psi < 144$, n = -3
×	$4 < \Psi < 13$, R > 400km	+	$167 < \Psi < 176$, R > 400km

arrives on a subsequent sweep at the same phase angle and apparent range as echoes from 38° elevation. A meteor at Range = 560 km and Phase = 330°, appears on the phase range diagram in the same place as one at Range = 150 km and Phase = 330°. About half the meteors with relative phase angles between 310° and 350° represent contamination from meteors arriving on a subsequent sweep. For a meteor travelling north south and detected at 9° elevation the specular reflection points of each receiver site will at most be separated by 0.6 km. Even a slow 20 km.s^{-1}meteor particle will cover this distance in about 11 radar pulses. More typical velocities will produce timelags even smaller. The associated timelag measurement errors make any velocity determinations for low elevation echoes uncertain. The two groups of echoes need to be distinguished and the low elevation group of observations rejected.

If the slow meteor echo described above were assumed to come from 38° then the 11 pulse timelag would imply an 80 km.s^{-1}velocity. Very few hyperbolic meteors are detected. Assuming all meteor echoes with phase angles between 310° and 350° come from 38° elevation and rejecting any with a velocity greater than 80 km.s^{-1}will remove observations from ranges greater than 400 km in most cases. Slow low elevation meteors moving well away from the north south line may still be included. The number of meteors with phase angles between 310° to 350° is relatively small; less than 10% of the total.

4.3.3 Phase Calibration Constant

The relative phase of the two signals measured by the receivers is not the same phase angle as at the aerials. A phase calibration constant needs to be added to correct for the unequal lengths of transmission line from each of the aerials[15]. Fitting the data to a set of constant altitude curves on the phase range diagram allows this calibration constant to be estimated. This approach requires several minutes of processing time on the PC for each hour file inspected. The bulk of meteors detected should ionise slightly lower than 100 km. Particular attention was paid to the position of the line of points associated with the ambiguous range values. Echoes from the north and south should also provide a symmetric distribution around the 180° phase angle level.

Once the whole reduction package had been assembled it became more efficient to inspect the altitude distribution directly. It also became obvious that it was necessary to check and adjust the phase calibration constant to match different periods of observations (see 4.3.4 below). Meteors with ranges

[15]As from 1991 February 24 the layout of the open wire transmission lines was changed. The two lines were brought to a common point equidistant from the array feed points. From here two coax cables of approximately the same length run into the receiver hut. The new layout built to minimise the effect of wet wires (see section 4.3.4). A phase calibration constant is still necessary.

greater than 400 km had been eliminated for the reasons noted above. This removed the data to fit the 4°- 13° elevation curve. The speed with which a reduction run could be done as a batch job on the University VAX mainframe rather overshadowed running phase calibration programs on the PC. The number of points that could be readily included greatly increased, making the distribution more reliable.

The combination of radar frequency, transmitter power and aerial configuration should give a peak to the altitude distribution in the 95 to 100 km region. By adjusting the value of the phase calibration constant this distribution can be moved around. An altitude distribution which peaks at 97 km and has a fairly sharp cut-off above 100 km is the ideal. Where the data does not exhibit this distribution the value of the constant can be changed and the reduction run recalculated. The phase calibration constants for individual days are stored in the PhaseCal.Dat file. Some days have two or three different values for different weather conditions during the day.

Figure 4.17 compares the changes in the altitude distribution by adding or subtracting 20° from the phase calibration constant. Usually the value of the constant is fixed to within about 5° by this calibration technique. Adjusting up or down by 10° on the chosen value produces a distinct shift of the distribution outside the expected altitude range from 95 km to 100 km. The calibration constant, for a given day, determined from these altitude distributions agrees well with the value needed to fit the phase range diagram.

4.3.4 Wet Wires

Ideally the appropriate phase calibration constant should remain constant though possibly changing slowly by a small amount as the coaxial transmission lines weathered. Unfortunately this was not the case. The value needed to establish the altitude distribution and fit the phase range diagram changed by up to 20° or 30° from one night to another. The altitude profiles in Figure 4.17 shows a considerable difference with changes in the calibration constant of this magnitude.

The variation seemed to be related to differing weather conditions during the observing run. Figure 4.18 shows the value of the phase calibration constant with a note on the weather for that day. Until this problem became evident the weather log produced for each run at the station was rather fragmentary. Only those days with definite comments on the weather are included in the plot.

The signals from the two receiver aerial arrays are taken from the centre of the feed lines that run along them. Open wire transmission lines run from these points to the receiver hut. They are connected to a coaxial balun ($\frac{1}{4}$ λ back from end plates) which runs into the hut and to the receivers. The length of the two open wire sections are different for each receiver. The one from the north aerial is 45m longer than the one from the south aerial.

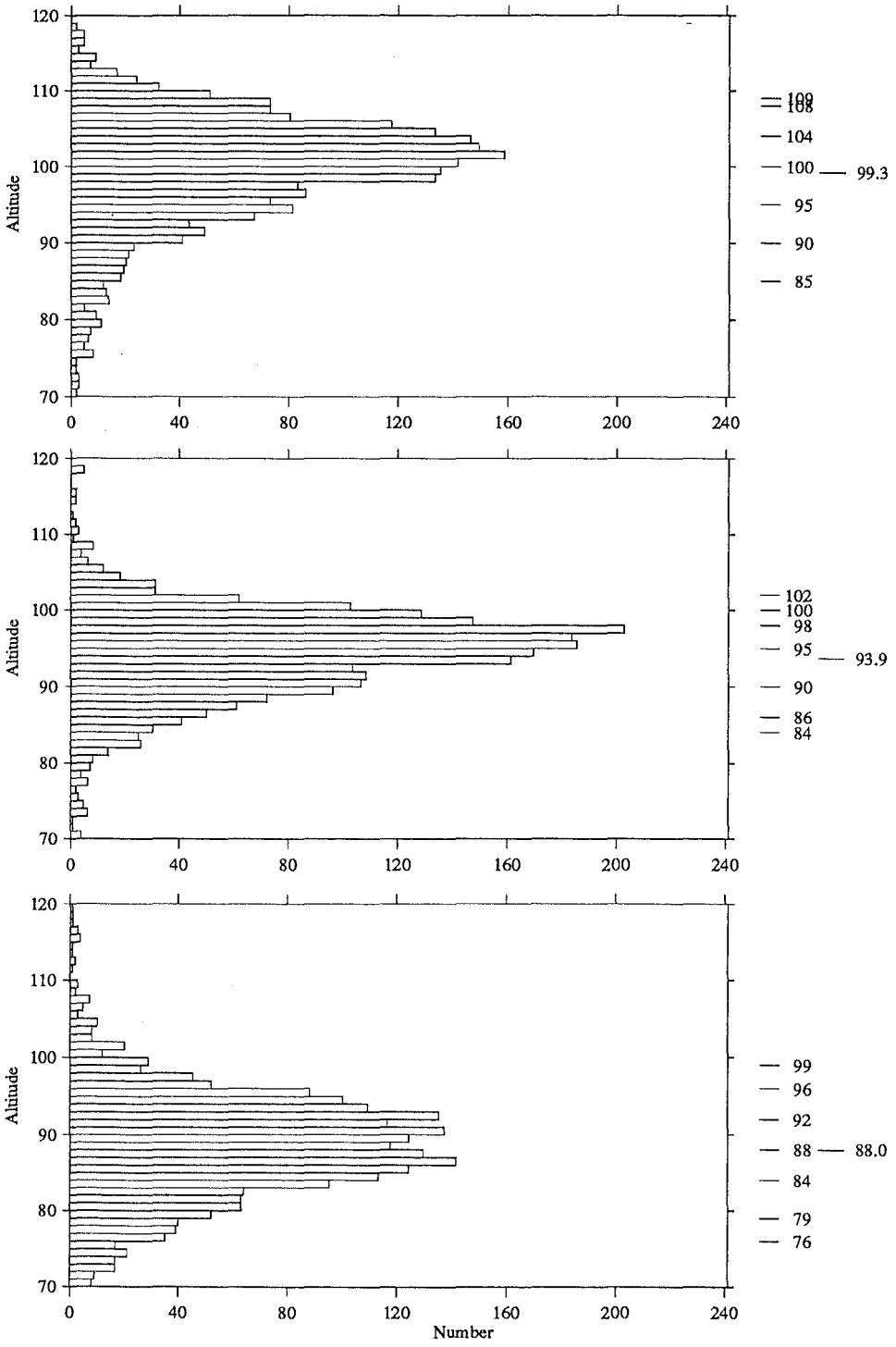


Figure 4.17: The altitude distribution for different values of the phase calibration constant. The 2244 observations plotted are from 1990 June 4. The centre plot shows the distribution given the constant finally chosen. The top plot is the result of using one 20° smaller producing a distribution skewed to higher altitudes. With 20° added to the calibration constant the bottom plot skews to lower altitudes.

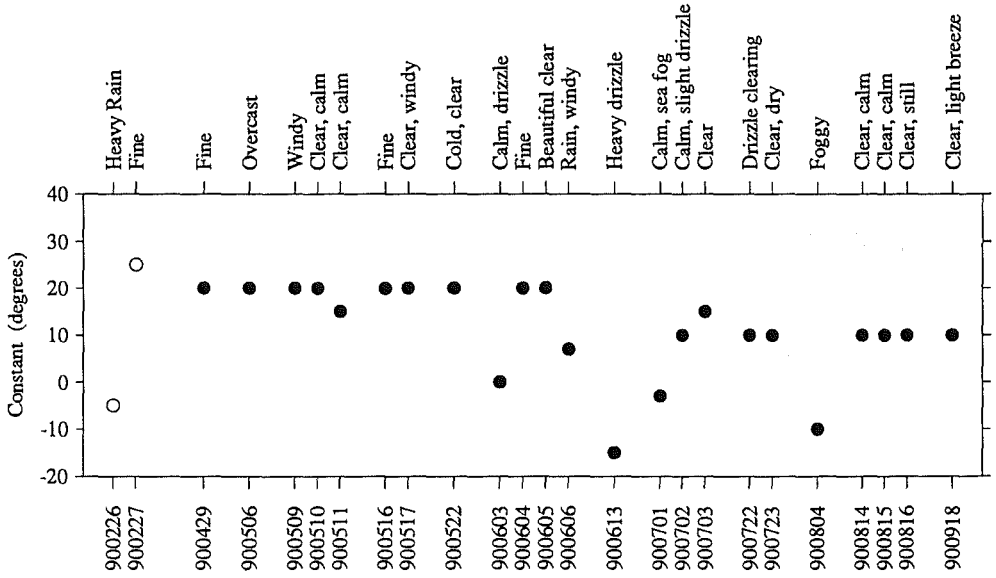


Figure 4.18: For observing runs where the log includes a note on the weather the value of the phase calibration constant is plotted. Prior to 1990 April the lengths of the two transmission lines were different. The open points from 900226 and 900227 actually have values of +165 and +195 respectively. They are plotted to compare adjacent days with different weather. The filled spots correspond to calibration constants from the scale on the left.

On days when the weather was dry the phase calibration constant remained relatively stable at around 20° (see Figure 4.18). Wet weather was associated with a constant significantly less than 20°. The larger excursions are on days with more moisture and less wind. The changes in the phase calibration constant seem to be associated with the amount of water sitting on the open wire transmission lines.

Moisture on the wire will act as a dielectric surrounding the conductor. The velocity factor for the transmission line will therefore change thus shortening the wavelength of the signal travelling along it. A signal from the north aerial will go through more cycles when it is wet compared with the south path. The relation between the phase difference measured at the receivers and the difference at the aerials will change. This is evident as a change in the phase calibration constant on wet days. More moisture around the wire increases the effective dielectric producing a greater change on wetter days.

This same effect has been observed before. Kwa (1977) made continuous measurements of 2.4 MHz antenna impedances in an effort to detect effects of tidal variations. During periods of rain the impedance was found to rise substantially. The effect was attributed to the transmission lines connecting the impedance meter to the antennas.

A shift of 20° in the value of the phase angle will produce a significant error in the observed velocity. This variation with wet wires needs to be allowed for. I have incorporated this into the reduction package by including a data file PhaseCal.Dat that contains the calibration constant to be used on any given day. This allows the data to be re-reduced without any manual intervention. As the wires dry out once the rain or fog ends the phase constant will change. To cope with this a different value of the constant can be used for each hour during the day. The start and finish hour needs to be noted with the value of the calibration constant used.

Over the period shown in Figure 4.18 the *baseline* of the phase calibration constant seems to have drifted by about 10°. The drift is of the same order as the uncertainty in both the determination of the calibration constant and individual phase angle measurements. The drift could be associated with weathering of the two sections of URM76 coaxial cable.

The changing moisture content on two different lengths of transmission wire introduces the largest uncertainty into the determination of the elevation angle. The whole situation was unsatisfactory requiring monitoring of data for each day. The extra losses involved in taking coaxial cable right out to the feed points would reduce sensitivity by about 2 dB. Coaxial cable outside in the salt air at Birdlings Flat (RX antennas about 500 m from the sea) does not survive that well. The best solution is to ensure that the change in effective path length with wet wires is the same for both transmission lines. The layout of the transmission lines from the Home receiver arrays was rearranged, 1991 February 20, to do this. Open wire lines were brought an equal distance to a common point close to the receiver hut. Equal lengths of coaxial cable now

provide the feed in to the receivers. The new phase calibration constant is 140°. This is noticeably less sensitive to how wet the transmission wires are.

4.3.5 Elevation Angle Uncertainties

This section provides a summary of the uncertainties involved in getting the elevation angle of the meteor trail. The radar provides a series of individual Tin and Tos values in the observation record. These are used to estimate mean Tin and Tos values. Where the standard error of the mean is greater than 5 A/D units the observation is abandoned. Mostly the uncertainty is significantly better.

The relative phase angle at the receivers is calculated from these mean values. There is some uncertainty where the fitting routine swaps from using one to using the other for determining the phase angle (see page 94). The first fit of this empirical relationship gave a 10° gap in the phase angle associated with these seams. The linear fits were adjusted to render this more nearly seamless (for observations on 4 June 1990). On some days small seams reappear. The problem remains with a best estimate of 10° uncertainty in the phase angle.

The relative phase angle at the receivers needs to be transferred to a measure of the phase difference at the aerials. This phase calibration constant is fitted to about 5°. The fit to the altitude distribution is usually better than 10°. The relative path difference to the two aerials is sensitive to moisture. Shifts in the constant of up to 30° are allowed for. Keeping the calibration constant accurate to 5° is possibly optimistic with wet transmission wires. For observations since 1991 February 20 this has been considerably improved.

A 10° uncertainty in the relative phase angle corresponds to a possible elevation angle error of 1°. For a typical elevation, $\Psi = 20°$ the velocity error from this 1° uncertainty is about 5%.

4.4 Timelags

The specular reflection points for spaced receivers will in general be at different points along the meteor trail. For the AMOR receiver stations these will usually be less than 3.5 km apart. The exact spacing for each individual meteor trail can be calculated from the station and echo geometry. By estimating the relative time-delay between the motion of the meteor through each of the specular reflection points, the atmospheric velocity of the meteor can be calculated. It is instructive to detail the procedures for determining timelag values that were investigated.

The following sections (4.4.1 to 4.4.5) discuss the various timelag measurement schemes. The differences between the position of the maximum slope on the rising edge section of the profile was finally chosen as giving the best estimate of the timelags. The observation displayed in Figure 4.19 is used as an example for the different schemes. Timelags have been calculated for this

900604 0308:17 #93, Figure 4.19	$t_1 - t_2$	$t_1 - t_3$	$t_2 - t_3$
Detection points (30,21,13)	21.0	-5.0	-26.0
Detection points (30,30,30)	14.0	-12.0	-26.0
Max points, smoothed (210,240,184)	23.0	3.0	-20.0
Rise point lags	22.1	-3.0	-25.1
Max rise slope lags	23.2	-2.9	-25.8
Full profile cross correlations	26.6	5.3	-20.2
Rising edge only cross correlations	21.8	-3.3	-25.0

Table 4.2: The timelags (in radar pulses) calculated for the 1990 June 4, 0308:17 #93 observation using the different techniques investigated. The parenthesis of the first three methods note the amplitudes used for each of the three channels. The maximum rising edge slope method was finally selected to determine timelags for the AMOR system.

observation using all the methods described. The results are summarised in Table 4.2.

A radar sweep is done every 2.641 ms. This gives a pulse rate frequency (prf) of 379 Hz. Timelags are calculated in terms of radar pulses. These can be converted to a time in seconds by dividing by the prf. The timelags within the computer data structures are called Lag12, Lag13 and Lag23 and defined Lag12 $= t_1 - t_2$. Where the time lag is negative the meteor echo occurred on the first named site first. Lag12 < 0 implies Home site detection occurred first followed by Nutt site.

Echo amplitudes for each radar pulse are temporarily stored in the computer memory. Where a radio meteor has been detected an amplitude profile of 250 consecutive sweeps from the appropriate range is recorded by the AMOR system. The amplitude of the meteor echo depends on the current length of the trail and the ionisation density at each point along it. As the meteor moves through the atmosphere the length of the trail increases producing an increase in the echo amplitude. The greatest increase in amplitude will occur just after the meteor moves through the specular reflection point. Most of the reflected echo strength comes from the region of the specular reflection point. Once the meteor has moved well past this point the echo amplitude of the meteor is dominated by the atmospheric influences on the ionisation density and orientation of the trail. Diffusion will reduce the density and the echo amplitude will drop away. Wind shear may bend the trail moving other sections of the ionisation column into a specular geometry.

The aim is to use the ionisation trail as a probe to measure the kinematics of the particle producing it. As noted above the rising edge of the amplitude profile is dominated by the particle's motion up to and through the

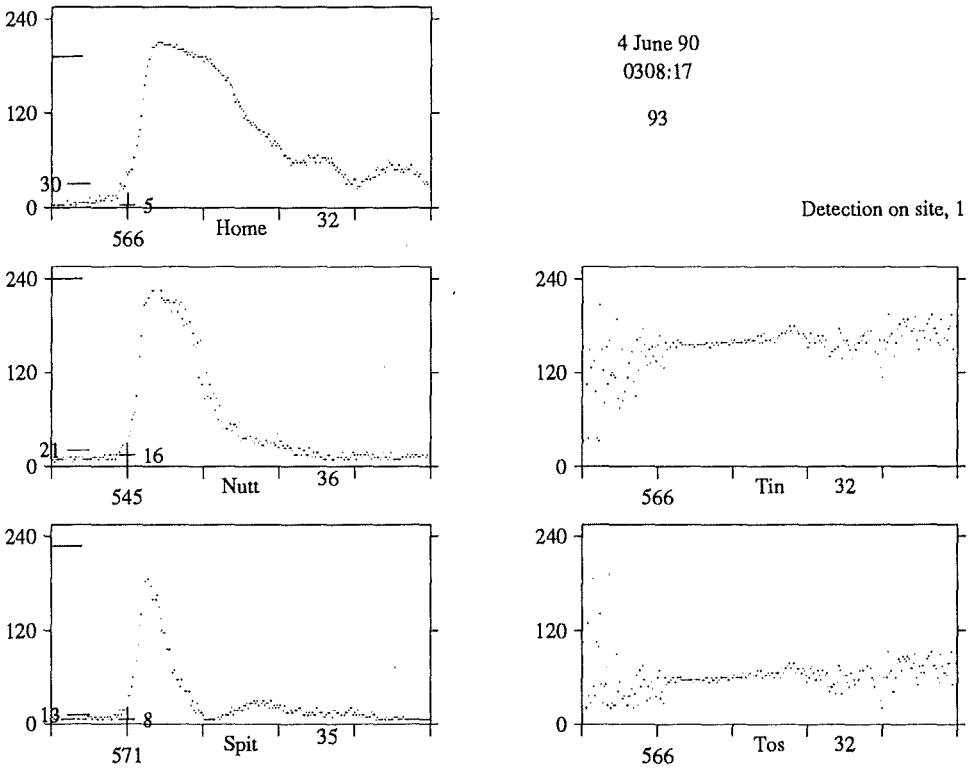


Figure 4.19: The raw observation used as an example for the different timelag calculation schemes.

specular reflection point. The decay slope is more related to the interaction of the ionisation with the atmosphere. To maintain reasonable accuracy in the observed meteor velocities, timelag measurement uncertainties need to be limited to about one pulse. Within the context of this section, a good timelag determination is one that agrees well with most of the alternative methods for determining them. Inspecting Table 4.2, the maximum points and full profile cross correlation methods perform poorly for the example observation. Where the timelag results are heavily effected by random noise or atmospheric effects the quality of the method is also diminished.

4.4.1 Detection Points

The observing software uses a very simple meteor detection algorithm. Once six consecutive echo amplitudes are recorded above the threshold level a meteor is identified at that range. Primary detection on the Home site has this level set at 30 A/D units. The remote sites use a level ten units above the current noise level for that channel[16]. In lifting the echo amplitude profiles out of the raw data these detection points are placed at index location 50. The sweep number associated with the detection point is also recorded in the site profile record. The reduction code calculates this basic difference in the timebase of the recorded echo profiles with the function LappLag.

The detection points of the three echo amplitude profiles in Figure 4.19 are all placed on the tick mark. The sweep numbers associated with these are 566, 545 and 571. The detection thresholds used are marked on the left of each profile. These detection points provide the first estimate of the relative time delays for the motion of the meteor past each of the different reflection points. Using 30 A/D units as the threshold for all three profiles gives a somewhat different set of detection points.

The detection points are not sufficiently accurate to be used as a general timelag scheme. Small variations in the ionisation density and random noise effects for low amplitude meteors noticeably affect the position of the detection point. Many large amplitude echoes exhibited an unexpected Fresnel-like interference effect as the meteor approached its specular reflection point[17]. Figure 4.20 gives an extreme example of this type. For a radar system where the receivers are not phase locked to the transmitter the amplitude is expected to rise steadily as the trail grows toward the specular reflection point. The interference affects the position of the detection point in a random manner.

[16]The threshold level went through a series of changes to enhance the performance of the remote site detection algorithm. It began as 30 A/D units and was changed to noise plus five units in 1990 June. Finally a detection threshold of noise plus 10 A/D units was implemented in 1990 August.

[17]We tend to describe profiles of this type as pre-Fresnel patterns.

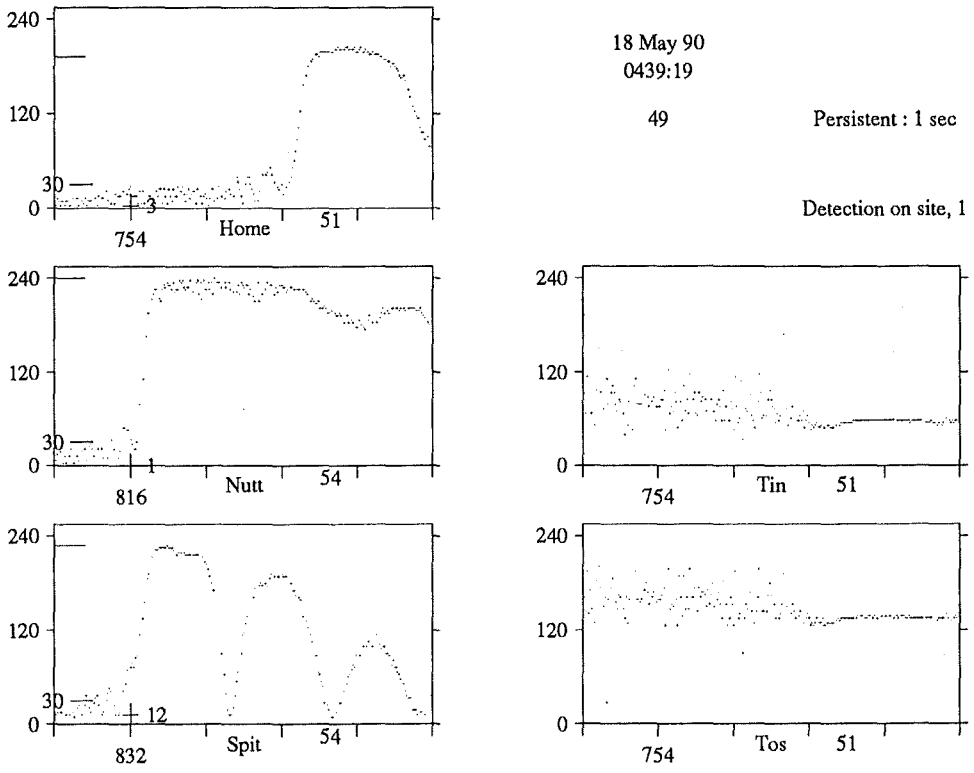


Figure 4.20: A number of large amplitude echo profiles exhibit a Fresnel-like interference pattern on their rising edges. This observation shows one of the more extreme examples of how this affects the position of the detection points. On the Home site the meteor was detected 100 sweeps before the main rising edge. The Nutt site was detected in the last oscillation while the Spit site detection occurred as the profile went up its main rising edge. Note that the recorded amplitude of the detection point on the Home is less that 30 A/D units as a result of the quarter, half, quarter averaging described in section 4.2.3.

4.4.2 Maximum Points

The position of the maximum amplitude in the profile provides another point that can be compared to provide relative timelags. These points represent the transitions between trail formation and the onset of decay. For a typical meteor the maximum echo amplitude will occur once the meteor has moved about half a Fresnel zone past the specular reflection point. This relative position of the meteor with respect to the individual specular reflection points is assumed to be the same for each receiver site geometry. For underdense echoes that show a rapid rise and subsequent exponential decay the location of the maximum amplitude of the profile gives a good estimate of the relative timelags. The location and size of the maximum amplitude is found using the FindMax procedure.

To reduce the influence of noise the location of the maximum amplitude is found using the smoothed profile. The location of the maximum for short duration small amplitude meteors is well defined and consequently gives good timelag estimates in these cases. For longer duration weak echoes small amplitude fluctuations tend to cause substantially different timelags.

Meteor echoes which saturate the receivers make it impossible to locate the position of the maximum amplitude. The long rise time for overdense meteor trails introduces uncertainty into the assumption that the meteor has reached the same relative position for each of the reflection geometries. The profile takes longer to reach maximum and hence the meteor will have travelled further past the specular point. The rise in echo amplitude is differentially affected by the different atmospheric conditions at each of the reflection points. The echo amplitude for other more transitional types continues to rise after the meteor has moved through the region around the specular reflection point. Figure 4.21 shows an observation of this type where the rise time to maximum echo amplitude is more a function of the plasma physics in the trail than the kinematic motion of the meteoric particle.

4.4.3 Rise Points

The location of the initial detection point and the maximum echo amplitude identify points both prior to and after the particles motion through the specular reflection point. The two techniques suffer from using just one location. In addition each location represents a point too far from the reflection point for accuracy to be maintained in certain classes of echo. A similar strategy where the average timelags for several amplitude levels on the rising edge of the profile was developed.

The timelags associated with points at 50, 60, 70 and 80 percent of the maximum amplitude are calculated. At least three of these must agree to within two pulses of the average value for the timelag to be accepted. These points represent a series of locations that straddle the specular reflection point

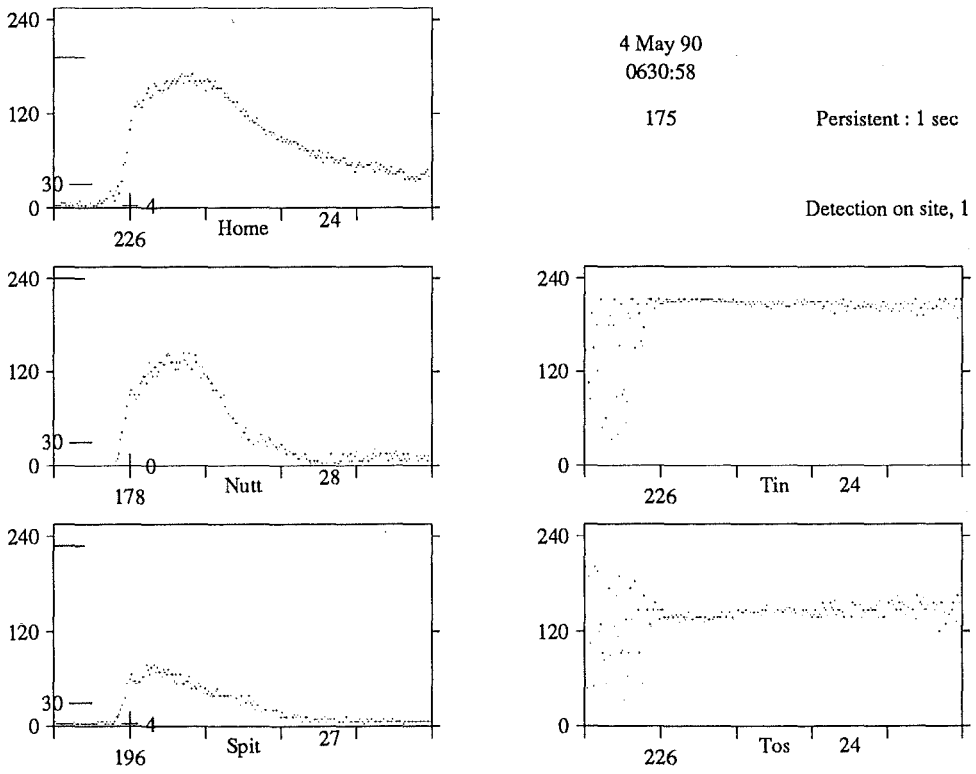


Figure 4.21: The echo profiles indicate a transitional (ie between under and overdense) ionisation density for the meteor trail recorded by this observation. The time it takes the profile to rise to maximum echo strength is strongly effected by the different atmospheric conditions and ionisation densities associated with each reflection point.

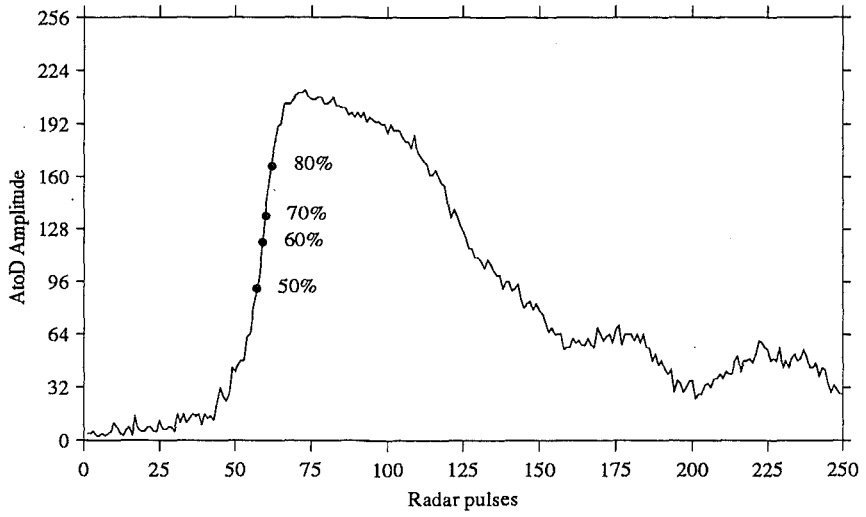


Figure 4.22: The rise points for the Home site profile of the timelag example observation are plotted as large spots. The relative time differences between these points on two profiles give four trial timelags. These timelags must agree reasonably and the average provides the rise point timelag.

in the classical radio reflection theory for underdense meteor trails. Figure 4.22 shows the locations of the rise points on the Home site profile of the timelags example, Figure 4.19.

The lower amplitude end begins above the onset of the interference patterns for most observations affected by pre-Fresnel oscillations, see Figure 4.20. Stopping at 80% of the maximum amplitude minimises the problems associated with meteor echoes saturating the receivers. Profiles that are affected by differences in the slow rise to maximum associated with transitional and overdense echoes can usually be identified automatically. Figure 4.21 is an example where the four timelags did not agree sufficiently for the timelags between the Home and Nutt channels and between Nutt and Spit to be calculated. No timelags could be determined for this observation.

The rise points scheme provides a relatively quick and easily programmed estimate of the timelag between the echo profiles. It suffers a little in accuracy since it uses relatively few points to estimate the timelag. The method does not work for observations similar to Figure 4.21 which have a well defined rising edge at slightly lower fractions of the maximum amplitude. The method does continue to provide timelags for some low amplitude echoes where the profiles look radically different. Where the shape of the rising edge on all three profiles is similar then the assumption that the kinematics of the meteor are dominating that section of the profile is strengthened. The rising edge on the

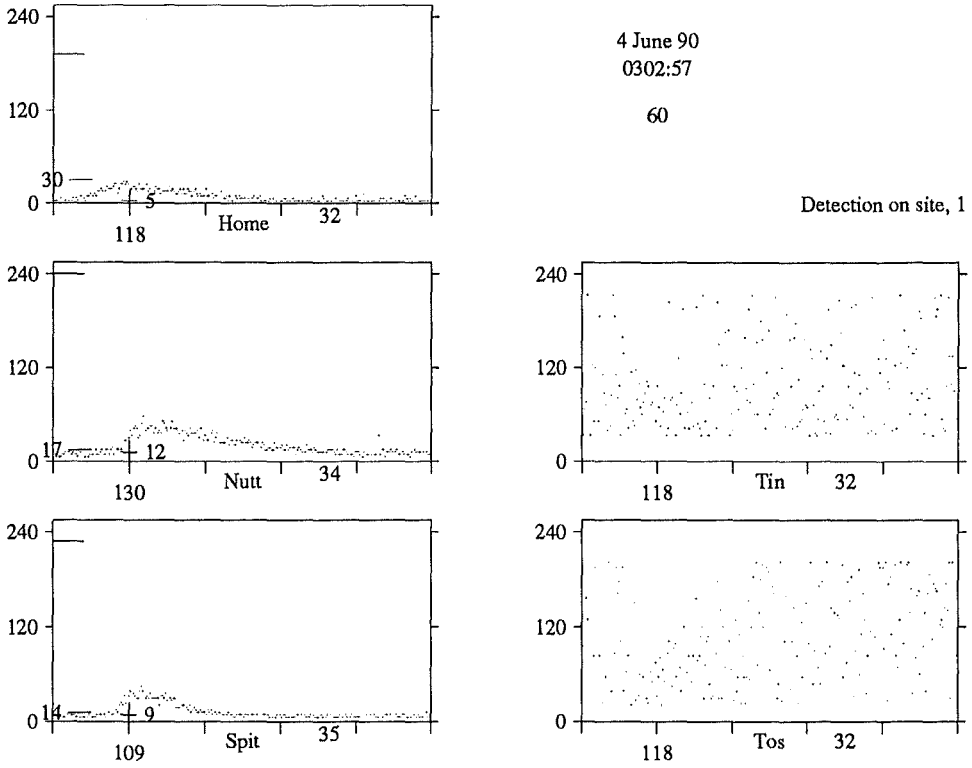


Figure 4.23: The rising edge of the Home site profile shows a noticeably different slope. It is probably associated with a wind shear deforming the ionisation trail into a reflection geometry. The relatively rapid drift in the phase data is consistent with the specular reflection point moving along the trail with the changing geometry.

Home site profile in Figure 4.23 is much shallower than that on the two remote sites. The rise in echo amplitude on the Home channel is more likely to be associated with wind shear deforming the trail into a reflection geometry. The phase record for the Home receiver station shows a fairly rapid drift that is consistent with this interpretation. Ideally the timelag determination routine would identify the different rising edge slope. The observation could then be rejected as not meeting the assumptions implicit in the AMOR system.

4.4.4 Maximum Rising Edge Slope

As the meteor moves through the specular reflection point the echo amplitude rises rapidly. Identifying the point at which the maximum rate of increase occurs on each of the profiles allows timelags to be estimated. The amplitude

profile is first differentiated numerically using the DifferentiateProfile procedure. The maximum slope is found from this differential by the FindMaxRise routine. A parabolic fit is used to improve accuracy.[18] The different timebases for the individual profiles are found using the LappLag function. Finally the timelags are calculated from this data, within the MaxRiseSlopeLags routine.

Figure 4.24 presents the diagnostics display for the maximum rising slope timelag calculation scheme. The top panel shows the first differential of the amplitude profile. The derivative is only calculated through to five pulses after the maximum amplitude recorded in the profile. The location of the maximum slope is marked on the plots of both the first derivative and the raw profile. A line showing the calculated maximum slope is overlaid on the amplitude profile. The two vertical lines show the rise time estimated for the meteor echo. They intersect with the maximum slope at the noise and maximum amplitude levels respectively.

The first differential is calculated numerically from the raw amplitude profile. The finite difference scheme comes from Comrie (1950). The derivatives were calculated based on the seven point scheme reproduced here.

$$wf'_0 = \frac{1}{60}(-f_{-3} + 9f_{-2} - 45f_{-1} + 45f_{+1} - 9f_{+2} + f_{+3}) \qquad (4.9)$$

To ensure that noise fluctuations did not influence the differential finite differences were calculated over a fairly wide timebase. A width of five ($w = 5$) was chosen. For an individual pulse labelled i the corresponding derivative for that sweep is therefore

$$f'_i = \frac{1}{5 \times 60}(-f_{i-15} + 9f_{i-10} - 45f_{i-5} + 45f_{i+5} - 9f_{i+10} + f_{i+15}) \qquad (4.10)$$

The parabolic fit used to locate the maximum rising edge slope uses the five points on either side of the maximum value in the derivative. This point is marked on the top panel as a plus symbol and on the amplitude profile as an arrow. The combination of five wide, seven point differentiation and a ten point parabolic fit means that the location of the maximum slope is fixed by at least twenty data points. Effectively it depends on echo data for about ten sweeps on either side of its calculated position. This greatly improves the accuracy. Problems with noise and interference spikes are virtually eliminated.

Some echo profiles do not have well defined rising edges. The Home profile of Figure 4.23 is such a case. A number of them show two distinct but displaced sections on their rising edge. For these observations it is not clear which of the two should be chosen. To try and eliminate these two conditions from further

[18]The location of the maximum could be found by calculating a second derivative. The maximum slope would then occur at the crossing point where this function equals zero. This was not tried because the routine to find the peak of a cross correlation function could be readily adapted to find the maximum of the first differential.

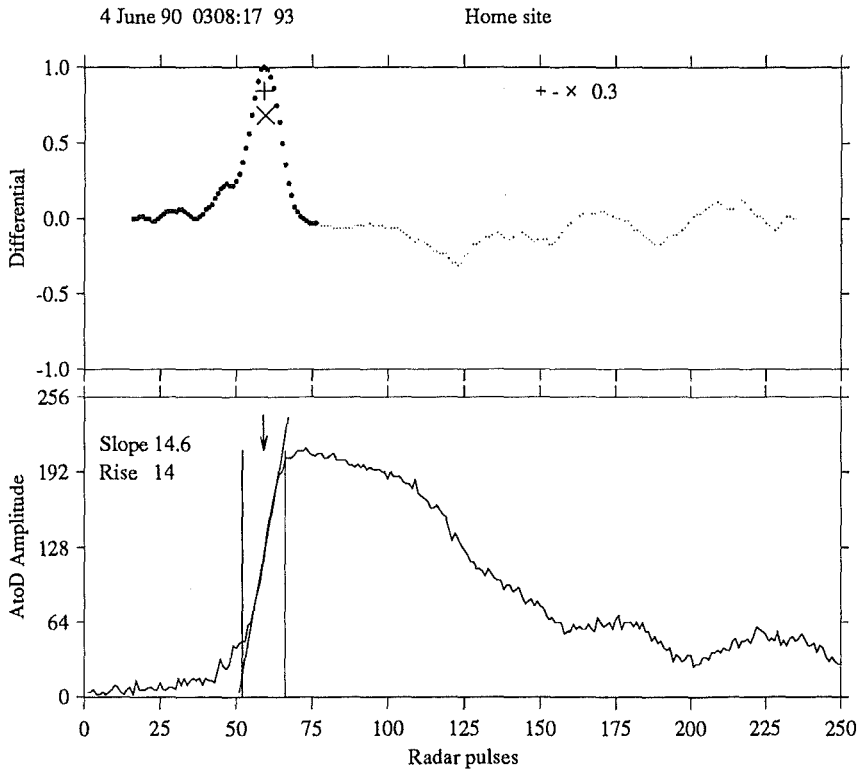


Figure 4.24: The amplitude profile and first differential used to locate the maximum rising edge slope. Within the timelag routines the differential is only calculated to just past the maximum amplitude; larger spots. The plus symbol is the result of a parabolic fit on the peak of the differential. The cross marks the point half way between slopes 70% less than the maximum.

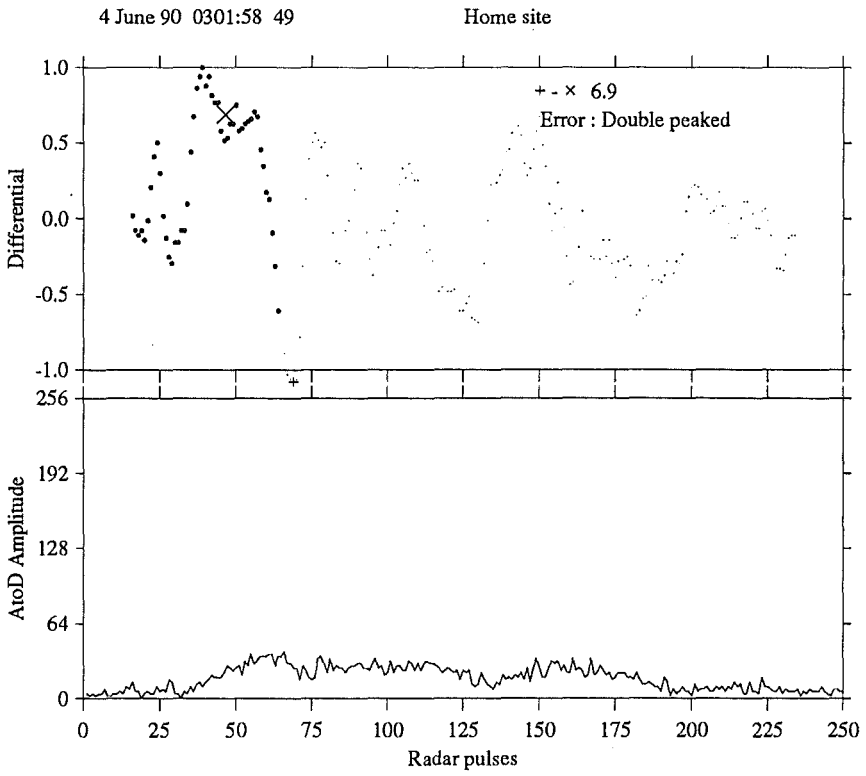


Figure 4.25: The Home site profile from Figure 4.23 does not show a well defined rising edge. The location of the maximum slope is too uncertain. Specifically the midpoint between the two points with 70% of the maximum slope is 6.9 pulses different from the calculated maximum slope. Unambiguous timelags cannot be calculated and further reduction is abandoned.

data reduction an alternate estimate of maximum slope is made. The extreme sweep locations where the slope falls below 70% of the maximum are found. A point halfway between these two is near the maximum slope for profiles where the point of maximum slope can be unambiguously established. The cross used to mark this point in the diagnostic graphics output is placed at 70% of the maximum positive slope. Where the two methods are more than 6 radar pulses apart the profile is assumed to have no clearly defined rising edge. In cases of this sort the derivative usually shows double peaks corresponding to two steeper sections to the rising slope. Figure 4.25 shows the profile and derivative for an example of this problem.

Data Screening Benefits

Echoes due to reflections off sporadic-E layers have echo profiles similar to those from meteor trails. These cannot be distinguished from meteor echoes by the detection scheme. Many of these sporadic-E triggered observations are rejected because they do not have concurrent *detections* from the remote site receivers. As the sporadic-E ionisation becomes more fully established in the radar beam, three station detections in similar range bins frequently occur. The amplitude level for sporadic-E is usually small and fluctuating. Frequently the 70% level estimate and the actual location of the maximum slope do not agree. To be passed for further reduction and hence accepted as a meteor observation, the condition needs to be met on all three profiles. This criterion almost completely eliminates contamination from sporadic-E triggered detections.

Echo Rise Times

Some additional information about the profile shape is also recorded by the MaxRiseSlopeLags routines. The value of the maximum slope is explicitly calculated within the routines. From this an estimate of the rise time for the echo is made. The location and magnitude of the maximum amplitude is found prior to the timelag determination. An estimate of the noise level is made for adjacent range bins at the time the meteor observation was recorded. The rise time is the number of pulses it would take an echo to rise from noise to the maximum amplitude given the maximum slope. The value of the maximum slope is not recorded directly in the reduced data record. It can always be recovered from the noise, maximum amplitude and rise time values that are recorded.

4.4.5 Cross Correlation

The relative proximity of the three reflection points initially lead to the hope that the three echo profiles would be reasonably identical, both in formation and subsequent evolution. The obvious way to find accurate timelags between similar amplitude profiles was by calculating a cross correlation function[19]. For short duration underdense meteor trails the profiles are certainly very similar. In fact for all the timelag schemes discussed this class of echo profiles gives reliable timelags. For less ideal meteor events the profiles are not always sufficiently similar to use the cross correlation method.

The different reflection points occur at different atmospheric densities. The echo profile associated with each station will therefore have different decay

[19] Considerable effort was expended on this technique before it was finally abandoned in favour of the maximum rising slope timelag approach. I believe the timelag cross correlation routine was the first piece of software written for AMOR meteor orbit reductions (in 1983). This qualifies as the first piece of software written and the last piece abandoned !

times. The ionisation density along the meteor trail varies as a function of atmospheric density as well as meteor mass and velocity. Different points will be associated with different diffusion regimes in addition to different maximum echo amplitudes. Wind shear, also, is a function of height and hence effects the reflection geometry for each of the receiver sites differently. As noted earlier, these problems caused by the atmospheric interactions with the meteor trail are most pronounced in the decay section of the profile.

For strong meteor echoes there was sufficient information in the rising edge of the profile to use just this section to calculate the station timelags using cross correlations. Meteor echoes that saturated the receivers still had undistorted rising edges and hence could have cross correlation timelags calculated for them. The rising edge cross correlations also have the advantage of removing from consideration the worst atmospheric distortions in the decay slope. However using just the rising edge to calculate the timelag makes it much more dependent on how the end points of the rising edge section are chosen. Low amplitude echoes are poorly served by just using the rising edge. The cross correlation timelag scheme uses the rising edge section in preference where all three profiles rise above 50 A/D units.

The cross correlation timelag between the Home and Spit sites of the observation displayed in Figure 4.19 is used as an example of the method. This is done using both the full profile and the rising edge section as a comparison. It is a situation where the full profile cross correlation performs poorly. The routine TimeLags which runs the cross correlations would have chosen the rising edge cross correlation in this case since the maximum echo amplitudes are large for all three profiles.

Full Profiles Cross Correlations

A cross correlation function (CCF) is calculated for any two profiles. The corresponding amplitudes from the two profiles are multiplied together. Each point in the CCF is the sum of these products. One profile is shifted by one radar sweep and the next point of the CCF calculated. Mathematically for two profiles, A and B, each point in the cross correlation function, CCF_j, is given by

$$\mathrm{CCF}_j = \sum_{i=1}^{250} A_i.B_{i+j} \tag{4.11}$$

The shift associated with the maximum value of the cross correlation function is the time displacement for the two profiles. A twenty point parabolic fit was used to accurately locate this maximum and hence calculate the timelag between the two profiles. Figure 4.26 plots the normalised cross correlation function for the example. Normally the CCF is only calculated out to a shift of ± 30 sweeps. This reduces the amount of calculation necessary. The method remains by far the most CPU intensive of those discussed.

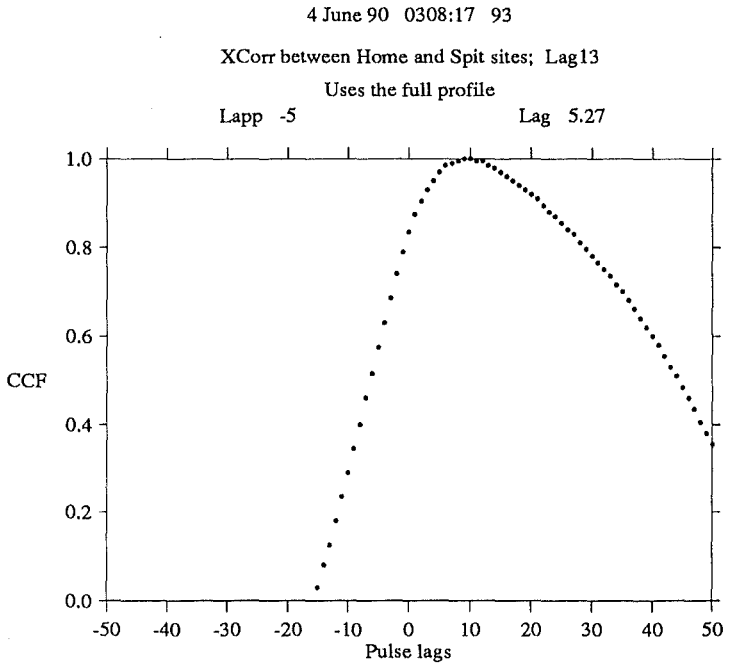


Figure 4.26: The cross correlation function for the timelag between the Home and Spit site profiles of Figure 4.19. The CCF has been normalised for display purposes only. The Lapp-lag, -5 is the difference between the two time bases. The relative shift calculated from the CCF is 10.27 pulses making the timelag Lag13 = 5.27 pulses. The skewed shape of the CCF is an indication of the dissimilarity of the two profiles.

Three steps were taken to enhance the accuracy of the cross correlation. The two, preferably identical, profiles rise from the radio noise to a maximum and then decay back to noise level. Where the meteor trail lasts for longer than the half second of recorded data the profile does not return to the noise level. This convolves a sampling window onto the profile which in turn effects the CCF. Recording 250 sweeps of data and placing the detection point at index 50 ensures these sampling windows are similar for all profiles.

As the CCF moves away from zero shift the extreme amplitudes of each profile are multiplied by zero. Their contribution to the CCF is lost[20]. Since the receiver continues to record noise values following the meteor echo it is more appropriate to multiply these ends by the noise level. It is easier to subtract the noise level from profile amplitudes. The CCF can become negative as is the case for shifts of -50 to -16 pulses in Figure 4.26.

The sensitivity of the CCF peak and hence the numerical accuracy of the cross correlation can be enhanced by subtracting the average amplitude of the profile. This is done after shifting the baseline to the quiet (noise) level.

The timelag between the Home and Spit sites for this observation gives a good example of the problem that plagued full profile cross correlations. The meteor echo from the Home site lasts at higher amplitude much longer than that on the Spit channel. Wind shear certainly affects the meteor trail, probably moving the specular reflection point for the Spit site toward lower ionisation densities whilst shifting the Home reflection point to higher densities[21]. Cross correlation timelags are biased toward the centre of the profiles. These centres occur later in the Home site profile than for the Spit site. In this case wind shear affects on the echo amplitude move the timelag by about 8 radar pulses from that indicated by the other methods.

Rising Edge Cross Correlations

Where the amplitude for all three channels rises above 50 A/D units a modified version of the cross correlation scheme is employed. Only the rising edge sections of the profiles are used. The timelags calculated from these sections are less affected by the sort of atmospheric influences that dominate the decay slope. As the amount of data included in the profiles decreases the cross correlations become more sensitive to the sampling window discussed above. Forty sweeps of data are used in the rising edge cross correlation. The end of this rising section is placed where the profile rises to 80% of the maximum amplitude.

Figure 4.27 plots the sections selected as the rising edges of the profiles. The similarity in the shape of these rising edge profiles is much more striking than the full profiles. The time displacement is more probably due to the kinematic motion of the meteor than is the case for the full profile timelags.

[20]This is the principal cause of problems with different sampling windows.

[21]The situation is similar to the problem of using two different size sampling windows.

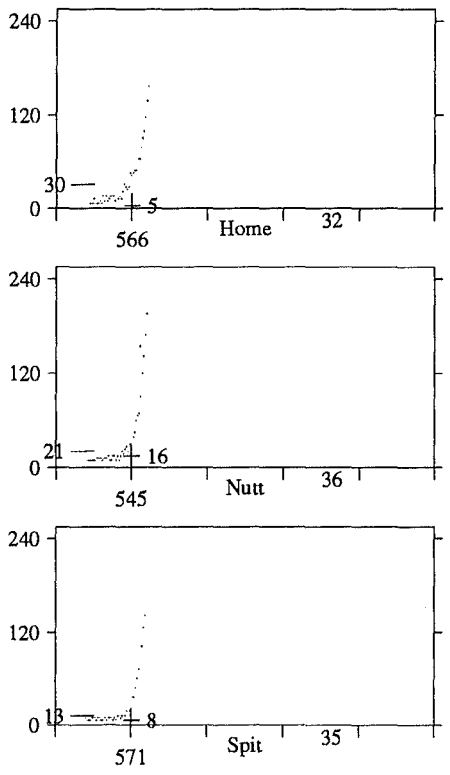


Figure 4.27: The rising edge sections used for the cross correlation timelag calculations. Each section contains the forty points through to 80% of the maximum echo amplitude. The full profiles are plotted in Figure 4.19.

The CCF for the Home and Spit site cross correlation is plotted in Figure 4.28. The shorter time series make the CCF much sharper than that using the full profiles.

In preference, the rising edge section should be used to determine the timelags within the cross correlation method. The scheme breaks down in a number of cases. The timelags from observations with dissimilar rising edge profiles are very sensitive to the data sweeps chosen as the rising edge. Where more of the large amplitude echoes are included in the rising edge section the CCF is skewed by the placement of the window function. Locating the end of the rising edge at 80% of the maximum amplitude runs into trouble with observations like that in Figure 4.21. Here the different ionisation densities at the each of the reflection points take different lengths of time to rise to a maximum. The rising edge sections identified by the FindRisingEdge routine for this observation are plotted in Figure 4.29.

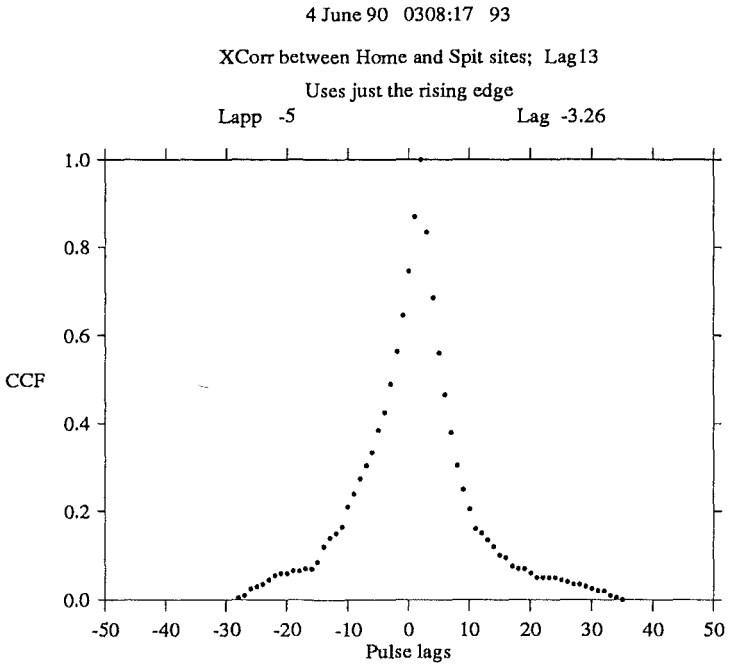


Figure 4.28: The rising edge cross correlation function for the Home and Spit site profiles of Figure 4.27. The Lapp-lag, -5 is the difference between the time bases of the two profiles. The relative shift calculated from the CCF is 1.74 pulses.

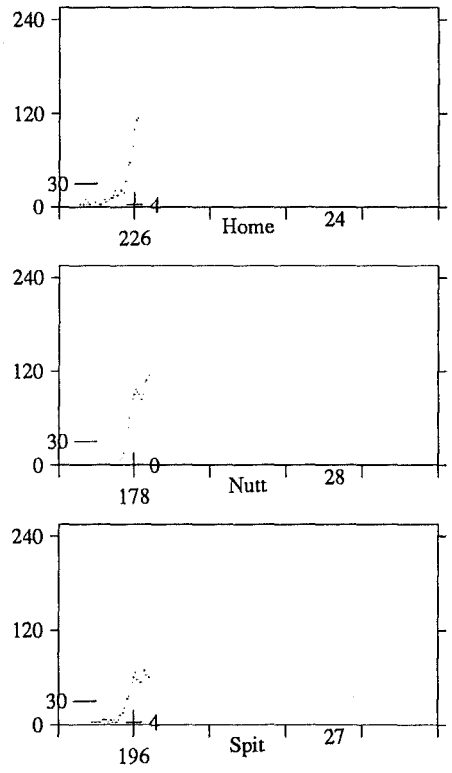


Figure 4.29: The rising edge sections identified by the GetRisingEdge routine lead to poor cross correlation timelags for the Figure 4.21 observation. The selection is very sensitive to the the location of the point with 80% of the maximum echo amplitude.

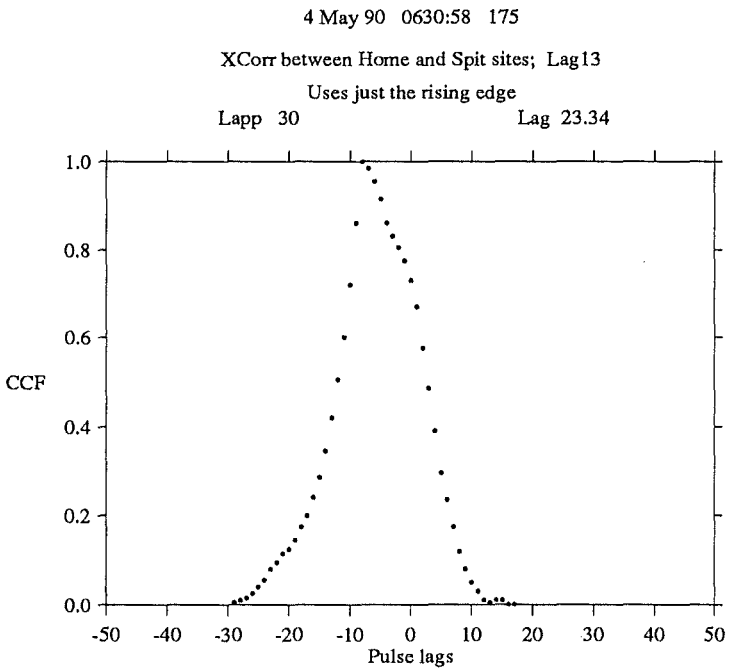


Figure 4.30: The CCF for the rising edge cross correlation timelag between the Home and Spit profiles of Figure 4.29. Timelags from this scheme are very sensitive to the position of the rising edge within the data profile. The CCF has been skewed away from the expected value as a consequence.

Figure 4.30 graphically demonstrates how the CCF can be skewed by dissimilar profile shapes. In this case differences in the two profiles were an artifact of the reduction process. For a satisfactory timelag reduction scheme the results should not be so sensitive[22]. Placing the sample window becomes increasingly difficult for echo profiles where the maximum amplitude is small. Overall putting the end at 80% seemed the best compromise for larger amplitude meteor echoes. The observation from Figure 4.21 is one example of where this was not entirely satisfactory.

Pre-Fresnel diffraction pattern echoes as in Figure 4.20 tend to produce rising edge sections of considerably different shape. The resulting cross correlation timelag often has more to do with these differences in shape than the motion of the meteor through the specular reflection points. The phenomena is often present on one profile and not the others thus exacerbating the problem.

Cross Correlation Timelags

To run the cross correlation scheme it was necessary to use both the full profile and rising edge section schemes. The choice of which one to use in any individual case always seemed rather *ad hoc*. Observations that fell on the borderline did not always give similar timelags for the two methods. To limit contamination of the results, the observation data records were viewed and sorted manually to eliminate those where the three profiles were not sufficiently similar for good cross correlation timelags. Typically about 20% of the observations with meteor echoes present on all three channels were rejected. In addition echoes off sporadic-E ionisation that appear on all three channels needed to be removed.

One attraction of the maximum rising edge slope timelag method was the more automated rejection of unsuitable observations. Screening by the MaxSlopeLags routine leaves about 4% contamination, nearly always wind blown echoes. This screening includes rejection of sporadic-E contamination. A manual sort is still done but this is greatly facilitated by having fewer observations and most of these *good*. On average 60% of all raw observations with detections made on the Home site are suitable for data reduction. The cross correlation scheme rendered good timelag determinations for the bulk of observations. These agreed well with the Fresnel velocity estimates. The stream membership and orbital distributions for the η Aquarid meteor shower that was used as a final calibration for the AMOR system gave a similar picture for both the cross correlation and maximum slope timelag schemes. The hope for fully automated data reduction, less *ad hoc* decisions within the code and greatly reduced CPU time prompted the change to the maximum slope method.

[22] In this case to the fairly arbitrary choice of 80% of the echo maximum as the end of the rising edge section.

4.4.6 Timelag Uncertainties

Much work was done trying to estimate the uncertainties associated with the cross correlation timelag calculations. This included generating artificial echo profiles, adding random noise and then modelling the data processing within the AMOR system. A known timelag could be introduced and the uncertainties estimated from a number of trials. The problems with unequal length data profiles were identified and eliminated during this investigation. Some consideration of the non uniform decay rates was included into this. Most simply this was done by allowing the ionisation diffusion rate at each reflection point to depend on the altitude and hence atmospheric density of those points. A one standard deviation error of one pulse covered the resulting timelags. For reasonably similar profiles a timelag uncertainty of one pulse is indicated. This should probably be stretched to 3 pulses for low amplitude echoes where the maximum falls under about 40 A/D units. Numerically, the cross correlation was considerably more accurate where the basic profile shapes were identical. The timelags are calculated to one tenth of a pulse.

The location of the maximum rising slope position is calculated to one tenth of a pulse. In a strict numerical sense the position is more reliable than this. The calculation considers a total of about twenty points centred on the rising edge of the echo. This compares with the amount of information used by the rising edge cross correlation scheme. A one standard deviation uncertainty in the AMOR timelags of one pulse seems justified.

The timelags measured by the reduction software are calculated from echo amplitudes. These are then assumed to represent the time-delay associated with the motion of the meteor particle through the geometric reflection points of the receiver sites. Most of the uncertainty in the timelags is associated with this assumption. Where the echo profile shape is the same for all three stations the assumption is most firmly justified. Atmospheric interactions with the ionisation trail certainly affect the decay section of the profile. For a few meteors with higher ionisation densities this interaction can extend to the rising edge. Noise dominates the very early part of the profile. The detection point marks where a definite radio echo has been received. Pre-Fresnel events also affect the early part of the rising edge of some larger meteor echoes. The maximum slope on the rising edge of the meteor profile lies between these features. Timelags calculated based on this point represents the best realisation of this basic assumption.

The one pulse estimate of the timelag uncertainties can be checked against the several alternate methods described in this chapter. Table 4.2 makes this comparison for the observation in Figure 4.19 used as the primary example of the various timelag schemes. Surveying this table (page 106) shows an agreement to better than one pulse for most of the timelags. The more severe disparities have been identified as problems within the individual timelag determination schemes.

900604 0308:17 #93, Figure 4.19	Lag12	Lag13	Lag23
Detection points (30,21,13)	21.0	-5.0	-26.0
Detection points (30,30,30)	14.0	-12.0	-26.0
Max points, smoothed (210,240,184)	23.0	3.0	-20.0
Rise point lags	22.1	-3.0	-25.1
Max rise slope lags	23.2	-2.9	-25.8
Full profile cross correlations	26.6	5.3	-20.2
Rising edge only cross correlations	21.8	-3.3	-25.0

900518 0439:19 #49, Figure 4.20	Lag12	Lag13	Lag23
Detection points (30,30,30)	-62.0	-78.0	-16.0
Detection points (30,30,30)	-63.0	-78.0	-15.0
Max points, smoothed (204,234,227)	24.0	43.0	19.0
Rise point lags	41.1	26.5	-14.6
Max rise slope lags	40.8	24.2	-16.5
Full profile cross correlations	999.9	999.9	-10.1
Rising edge only cross correlations	-80.1	-95.8	-15.7

900504 0630:58 #175, Figure 4.21	Lag12	Lag13	Lag23
Detection points (30,30,30)	48.0	30.0	-18.0
Detection points (30,30,30)	48.0	28.0	-20.0
Max points, smoothed (167,139,74)	57.0	48.0	-9.0
Rise point lags	999.9	33.1	999.9
Max rise slope lags	51.0	33.0	-18.0
Full profile cross correlations	52.0	35.4	-16.3
Rising edge only cross correlations	40.1	23.3	-16.9

900604 0302:57 #60, Figure 4.23	Lag12	Lag13	Lag23
Detection points (30,17,14)	-12.0	9.0	21.0
Detection points (30,30,30)	-17.0	4.0	21.0
Max points, smoothed (25,45,38)	-38.0	-3.0	35.0
Rise point lags	-30.5	-6.5	23.9
Max rise slope lags	999.9	999.9	999.9
Full profile cross correlations	-31.8	-2.6	27.3
Rising edge only cross correlations	-26.4	-4.6	22.0

Table 4.3: A comparison of the timelags for all the observations used as examples within this section. The timelag LagIJ is defined as $t_I - t_J$. The second of the two detection point lags for the two observations #49 and #175 uses the recorded echo profile that includes the quarter, half, quarter averaging of the observing code. The first uses the detection made while observing with the raw echo amplitudes. Routines that could not establish a timelag return 999.9 to indicate an error condition. For the rising edge cross correlation on observation #49 the GetRiseSection routine did not look far enough down the profile thereby producing wildly divergent results.

Additional comparisons for all of the observation records used as examples in this section are contained in Table 4.3. Where the individual timelags were identified as reliable given the method and echo profiles concerned, these results agree to within about one radar pulse. A more exhaustive data set has also been reduced and inspected. Where a timelag was determined using the maximum rising slope method it agreed to within one or two pulses of timelags calculated by other methods considered reliable given the echo profiles. The one pulse timelag uncertainty adopted for AMOR system seems justified in comparison with the other methods tried.

The meteor velocities calculated from the maximum rise slope timelags can be compared with Fresnel based velocity measurements in Chapter 5. A one standard deviation uncertainty in the timelag velocities of 2.5km.s^{-1} is indicated. For a typical meteor with a 30km.s^{-1} velocity observed at 25° elevation this uncertainty corresponds to a timelag uncertainty of approximately one pulse.

The 1990 observations of the η Aquarid meteor shower have been used as an astronomical calibration for the AMOR system. This is discussed in Chapter 7. The implications of measurement uncertainties on the distribution of orbital elements for stream orbits was investigated (page 215). The station parameters for a typical stream orbit were varied within a Gaussian distribution. The resulting distributions of orbital elements were similar to the observed distributions. A one pulse timelag uncertainty was used in this study lending confidence to its validity.

4.5 Meteor Diffusion Heights

Meteor echoes generally show a rapid rise in echo strength. This is associated with the formation of the ionisation trail in the region of the specular reflection point. This ionisation then diffuses away into the neutral atmosphere. As the cross section dimension of the column approaches the radar wavelength interference effects severely reduce the amplitude of the radio reflections. The echo strength drops away until it once again is dominated by sky noise. The diffusion rate can be calculated from the exponential decay slope of the profile. The diffusion and atmospheric density are related and from this a height for the meteor trail can be inferred.

For meteor trails with relatively diffuse ionisation columns the decay is dominated by ambipolar diffusion. In these cases the profile will show an exponential decay. McKinley (1961) outlines the theory for these underdense trails where the entire cross section of the trail contributes to the echo strength. The transition to overdense trails comes at about 2.4×10^{14} electrons/m. The exponential decay time, τ, for an echo profile is defined as the time it takes the echo amplitude to fall to $1/e$ of the initial amplitude. For an underdense

meteor trail this is related to the ionisation diffusion rate, D.

$$\tau = \frac{\lambda^2}{16\pi^2 D} \tag{4.12}$$

This diffusion rate is a function of the atmospheric density at the reflection point. Greenhow and Neufeld (1955) found a linear relationship between height, h, and $\ln D$. The slope is the scale height for the atmosphere in the meteor region. Jones and Jones (1990) provide a more recent empirical relation for the diffusion coefficient and height.

$$\log_{10} D = 0.06\, h - 4.74 \tag{4.13}$$

Equations (4.12) and (4.13) can be combined to compute the implied height of the meteor trail given the echo decay time. This is done for a decay time, τ, measured in pulses and a radar wavelength of 11.45 m. The diffusion height is calculated in kilometres. The equation is also rearranged to use natural logarithms which can be calculated by a Pascal function within the code.

$$\begin{aligned} h &= 7.24 \ln D + 79.0 \\ &= 79.0 - 7.24 \ln\left(\frac{\tau}{314.6}\right) \end{aligned} \tag{4.14}$$

In practice the observed echo decay rate is often affected by wind shear. The trail is bent by the wind profile moving the echo point up or down the column. The atmospheric density of the reflection point changes during the echo decay. The time since the ionisation was produced also changes. These factors alter the observed decay rate. Frequently the wind shear bends the trail in such a way that two points meet the reflection criterion. The echoes from the two sources interfere producing an oscillating amplitude pattern.

4.5.1 Echo Decay Time

Measurements of the echo decay time were originally made so that diffusion heights could be established. These can be compared with the geometric heights calculated from the echo elevation and range thus providing an independent check on the success of the elevation angle measurements. The theory discussed above pictures an underdense meteor trail where the decay is dominated by ambipolar diffusion. In such a case the echo decay is exponential. Care needs to be taken to ensure that the measured decay times are only attempted on profiles which show an exponential decay. If possible, diffusion heights should not be calculated for profiles affected by wind shear.

Figure 4.31 shows an example of the decay time determination for a profile that meets the requirements of the theory. The echo profile rises rapidly to a maximum and then exponentially decays away to noise. The amplitude profile is differentiated using the same scheme as the maximum rising slope timelag determination that is described in section 4.4.4. Only points starting from the

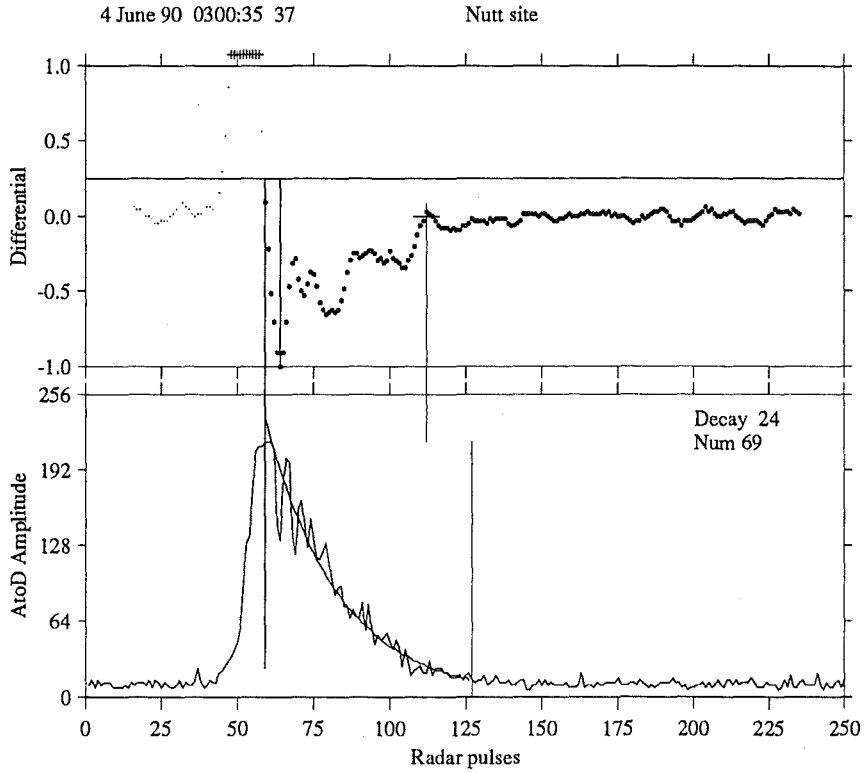


Figure 4.31: The amplitude profile and first derivative used in calculating the echo decay time for an example profile are plotted. Only the large spots of the derivative are calculated within the GetDecay routine. Starting from a point five pulses before the maximum decay slope a least squares fit of the exponential decay is done. The resulting curve is overlaid on the amplitude profile. This continues to include data points until the profile amplitude drops to noise plus 5 A/D units. 69 points are included in the fit and a $1/e$ decay time of 24 pulses calculated.

location of the profile maximum are calculated. These are plotted as larger spots.

A least squares fit assuming an exponential decay is carried out to find the echo decay time. This fit starts at a point five sweeps before the maximum decay slope of the profile. The derivative is then searched through to find the first slightly positive slope. Particularly strong Fresnel oscillations can cause this condition to be met very early in the profile decay. From this point the routine continues through until the amplitude drops down to noise plus 5 A/D units. Only the data between these two end points are used for the fit. The points are marked on the profile plotted in Figure 4.31.

Where there are less than 15 data points to be included, the fit is abandoned. To ensure reasonable numerical stability the data profile must record an amplitude drop of at least $1/e$. Recorded amplitude profiles are only 250 sweeps long. The decay section is always formed by less than 200 sweeps so only reasonably rapid decays will be detected. Very few decay times exceeding 100 pulses are found. Decay times of less than 5 pulses could not sensibly be found using this routine. The numerical differentiation is not sufficiently sensitive to do this. Such short decays are rejected.

The maximum slope for an exponential decay will occur at the start of such a decay slope. A meteor trail that begins as a transitional or overdense column will expand in time. Once the ionisation becomes sufficiently diffuse the radio echoes will move into the underdense regime. The echo strength will begin to fall off both more rapidly and exponentially. Starting the curve fit around the maximum decay slope gives the best likelihood of only fitting to the exponential curve.

Where the peak echo strength begins to saturate the receiver the recorded echo amplitude becomes non-linear (see 4.2.3, page 86). Figure 4.32 gives just such an example. Effectively the recorded decay does not become exponential until around index 175. Fitting an exponential decay to all the data after the occurrence of the profile maximum would not be sensible here. Additionally this example shows evidence of more complete receiver saturation around the peak echo amplitudes. Waiting for the maximum decay slope avoids problems with saturation in determining the decay times.[23]

Meteor trails that are unaffected by wind shear or uneven ionisation distributions will follow an exponential decay back to the noise. If a profile shows moderate evidence of a rising edge slope after the maximum decay point then the decay time fit is abandoned. Where there is a positive slope that is more than 25% of the steepest decay a comparatively sharp rise is identified. If the derivative rises above the +0.25 line the rejection criterion is met. For lower amplitude echoes this is more likely to occur. It effectively provides a

[23]It would be possible to convert these recorded A/D amplitudes to an implied peak amplitude for the meteor profile. This has not been done. Conducting this sort of calculation would give another perspective on the non-linearity introduced by the quarter, half, quarter average used to record the data.

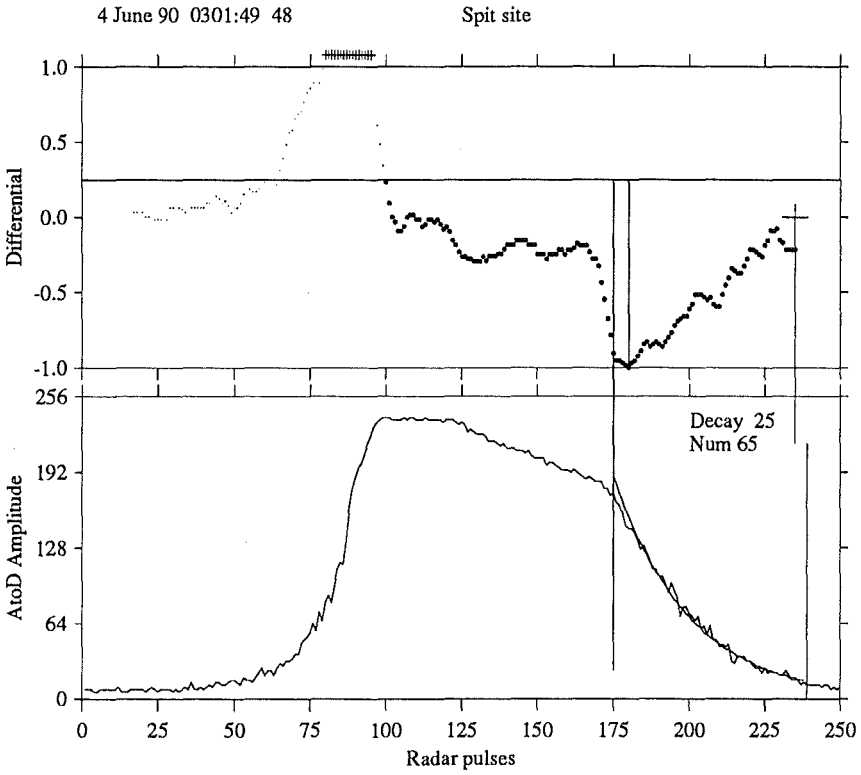


Figure 4.32: The onset of an exponential decay is found later in this echo profile. The least squares fit to find the echo decay time is begun five pulses before the maximum decay slope. Starting the decay fit here also avoids problems with receiver saturation. The maximum decay slope point effectively locates the onset of exponential decay.

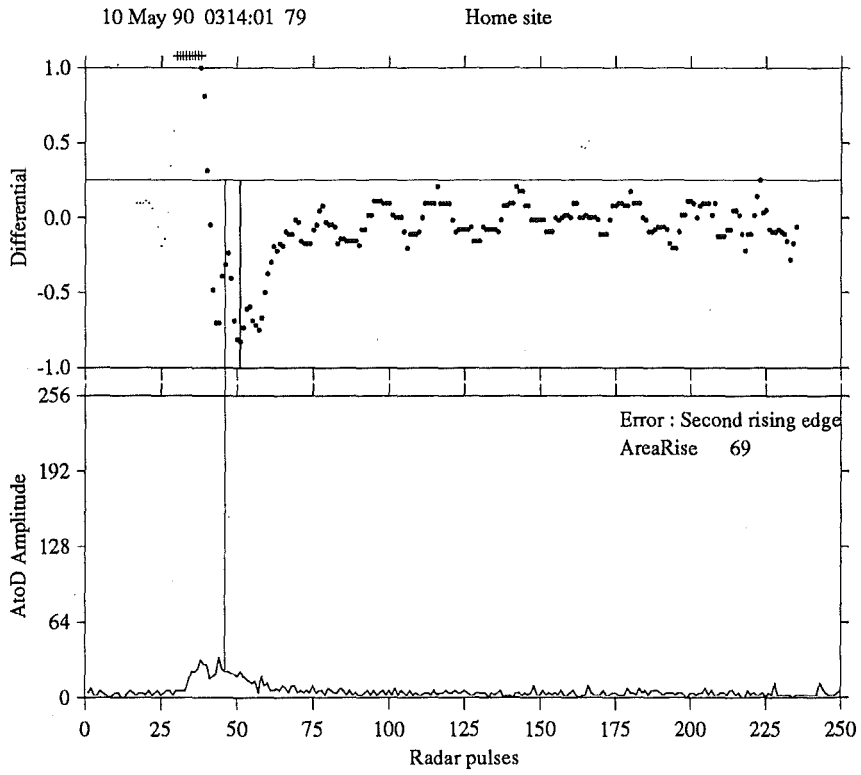


Figure 4.33: Where a rising edge slope numerically greater than the 25% of the maximum decay slope is found after the start of the decay section of the profile, no estimate of the exponential decay time is made. In this case the signal to noise is too low for a good characterisation of the decay. The derivative rises above +25% at around index 225.

low signal to noise cutoff for the decay time determination. Figure 4.33 shows such a situation.

Beating between echoes from two sections of the same meteor trail often produces less extreme rising edges. The phenomena is associated with larger amplitude echoes and consequently stronger maximum decay slopes. Profiles with longer less pronounced rising slopes in the decay section need to be identified. If any positive area under the derivative is greater than 20 A/D units then a consistent rising slope is noted. This is equivalent to a positive slope of 1 A/D unit per pulse for twenty consecutive sweeps. Figure 4.34 gives one such example. The oscillations of the echo decay slope are most probably the result of wind shear. No exponential decay time is measured and hence no estimate of the diffusion height attempted.

Echo amplitudes transmitted from the remote sites affected the base level of the video output on each radar sweep. Because of inadequate D.C. restora-

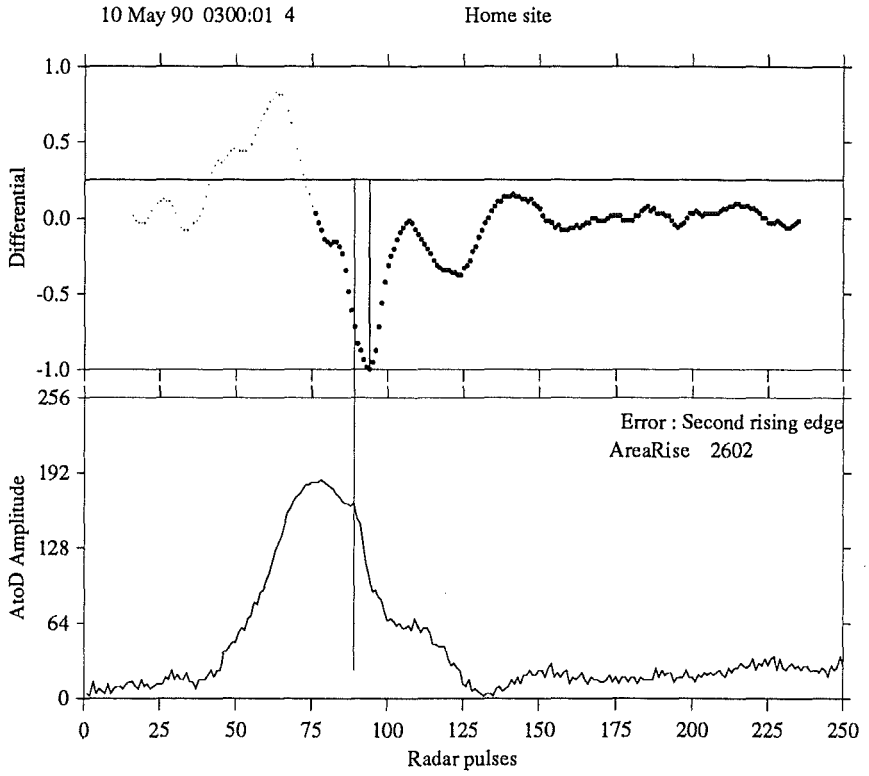


Figure 4.34: Profiles with consistent rising edges after the maximum echo amplitude do not meet the requirements of the underdense diffusion theory. Around index 140 the positive area under the first derivative is greater than 20 A/D units. The differential is evaluated without dividing by the 5 × 60 factor of equation (4.10). The AreaRise should be divided by 120 to give units of (A/D) / (pulses) × (pulses).

tion the baseline noise from the FM link receiver is pulled up when a strong echo is present (see page 80). This is most pronounced on the Nutt channel link. The stronger echo amplitudes are artificially raised compared with those lower amplitude echoes as the profile drops to noise. The decay times for the remote site profiles are larger as a result of this problem with the links. I had hoped to make comparisons between the diffusion heights for the three different reflection points. This is not possible without better clamping of the remote site baselines. Small changes in the more severe decay constants make a large difference in diffusion heights[24]. The situation was considerably improved in 1990 December when a clamping circuit was added to the video outputs of the remote sites. This was done to compensate for temperature induced long term drifts in the FM baseline. It had the additional benefit of reducing *baseline pull-up* on the remote site channels.

4.5.2 Comparison With Geometric Heights

A scheme for the measurement of the echo decay time was originally set up to provide a comparison for the geometric and diffusion heights of the AMOR meteors. This provides a check on the elevation angles determined from phase comparisons. The decay time is not necessary for the determination of meteor orbits. The distribution of diffusion heights for 1990 May 6 is plotted in Figure 4.35. This peaks at around 97 km. The phase calibration constant is chosen to place the peak of the geometric meteor altitude distribution at about this level. The geometric altitude pattern in Figure 4.17 provides a comparison. The drop off in the number of measured diffusion heights away from the mean is a lot sharper than the altitude distribution of the meteors detected (Figure 4.17).

The actual shape of the diffusion height distribution in Figure 4.35 is the product of various selection effects. The atmospheric density at 110 km means meteor trails diffuse in a very few pulses. A five pulse decay time corresponds to a height of 109 km. The AMOR radar pulse rate effectively limits the maximum diffusion heights. For higher altitude meteors the very rapid diffusion almost immediately makes the ionisation column of similar dimensions to the radar wavelength. Effectively the rapid diffusion produces a radar echo ceiling for a given radar frequency.

Fixed length echo profiles of 250 points are recorded by the system. Less than 200 points are recorded in the decay section of the echo. This places a limit on the echo decay time. A decay time of 200 pulses corresponds to

[24]This problem with slight baseline shifts on very large amplitude echoes does not seem to affect the timelag determinations. The maximum rising slope points are usually about halfway up the rising edge where the amplitude is still relatively small. Assuming the amount of baseline lift is proportional to the echo amplitude the position of the maximum rising slope will not be changed anyway.

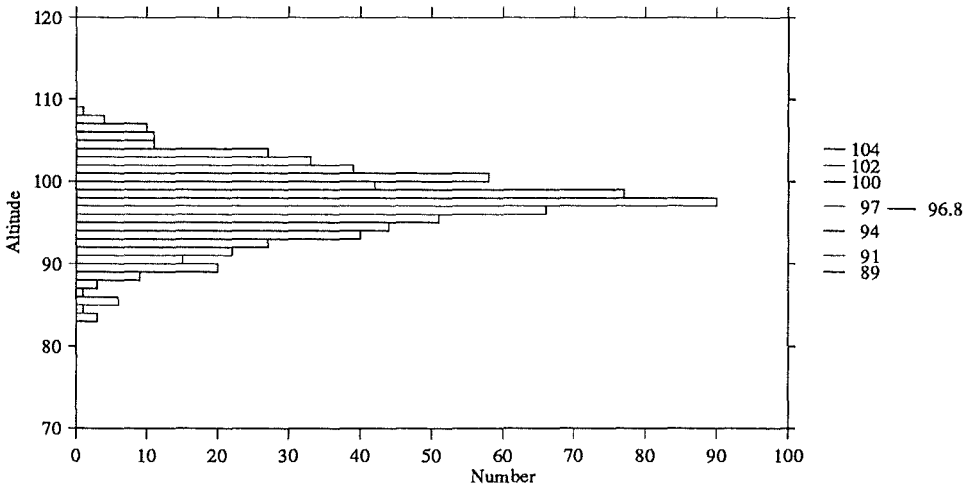


Figure 4.35: The cumulative distribution of the Home site diffusion heights for 1990 May 6. The distribution can be compared with the geometric altitude pattern in Figure 4.17. Measuring echo decay times of less than five pulses are considered calculation errors. This produces a high altitude cutoff at 109 km in the distribution. Profiles must show a decay of at least $1/e$ before the decay time is considered reliable. A decay of 100 pulses produces a low altitude cutoff at 87 km.

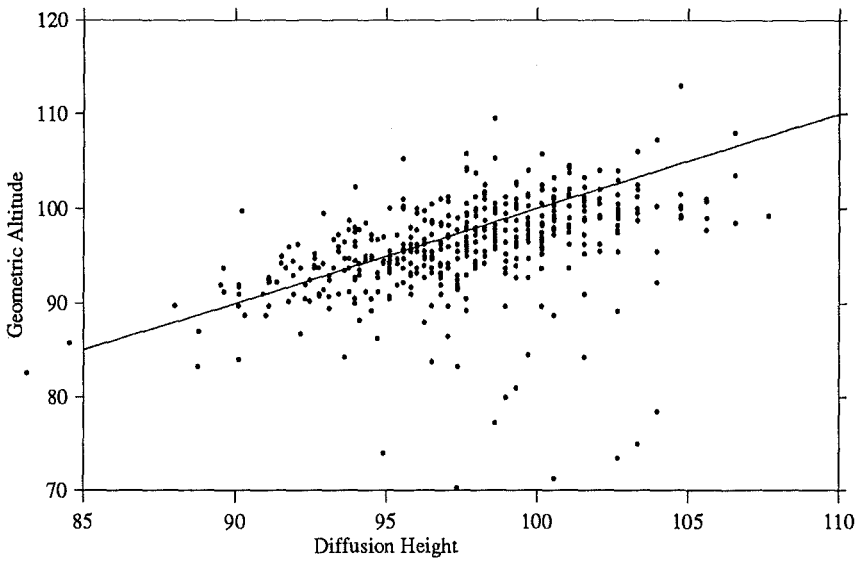


Figure 4.36: The geometric heights of the meteor trails are compared with the heights inferred from the diffusion coefficient. 474 comparisons of the Home site diffusion heights on 1990 May 6 are made. The line represents the set of points where the geometric height is the same as the diffusion height. The echo decay time values are only calculated as accurate to the nearest pulse. The lines in the data points that become evident as the diffusion heights reach higher altitudes are a result of this.

diffusion height of 82 km[25]. More usually meteor trails that last this long are effected by wind shear. In these cases exponential decay times are not calculated and hence diffusion heights cannot be inferred.

Figure 4.36 plots a comparison of the geometric altitudes with the Home site diffusion heights. The line plotted represents the set of points where the two heights are equal, rather than a best fit. Overall the two different heights agree well. Individual points show considerable scatter. Copious measurements by other radars have shown a large spread in diffusion for a given height. Rice and Forsyth (1963) summarised the suggested sources for this spread, see Howick (1991) for a recent discussion.

[25] Given the empirical relation of equation (4.13) it is mathematically improbable that any diffusion heights would be detected below 79.0 km !

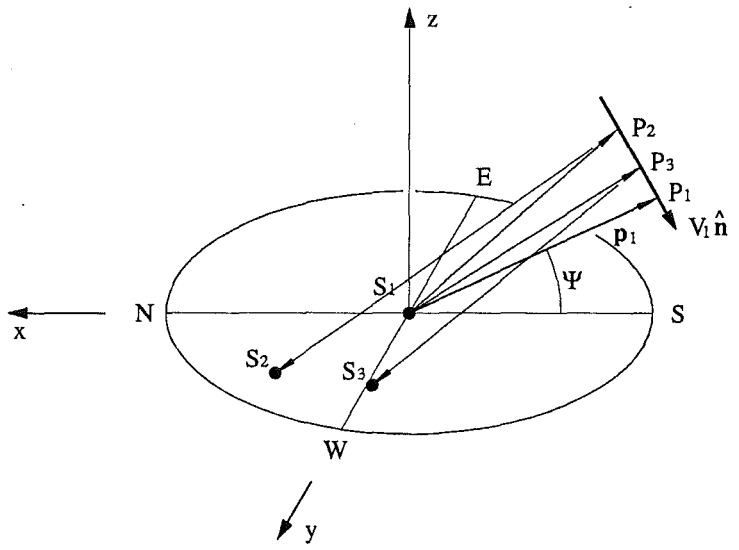


Figure 4.37: The reflection geometry for a meteor echo in a topocentric coordinate frame centred on the Birdlings Flat home station.

4.6 Azimuth, Zenith and Velocity

4.6.1 Observed Velocity Vector

Given the range, elevation and timelags for the three radar echoes the observed velocity vector of the meteor can be determined. Previous methods of derivation have been in terms of spherical trigonometry. The derivation used here employs a new method using vector expressions to describe the reflection geometry. The argument relies on two geometrically equivalent descriptions of the radar pulse reflection by the ionisation trail. Two vectors are perpendicular when their dot product is zero. More traditionally, the minimum distance from a point to a line will hit the line at right angles.

Using the coordinate reference frame described in Figure 4.37, the point in space on the meteor path which reflects the radar beam back to the Home site may be described by $\mathbf{p}_1$ where the range and elevation angle are given by R and Ψ respectively.

$$\mathbf{p}_1 = \begin{pmatrix} -R\cos\Psi \\ 0 \\ R\sin\Psi \end{pmatrix} \tag{4.15}$$

In the same topocentric coordinate frame the direction vector of the ve-

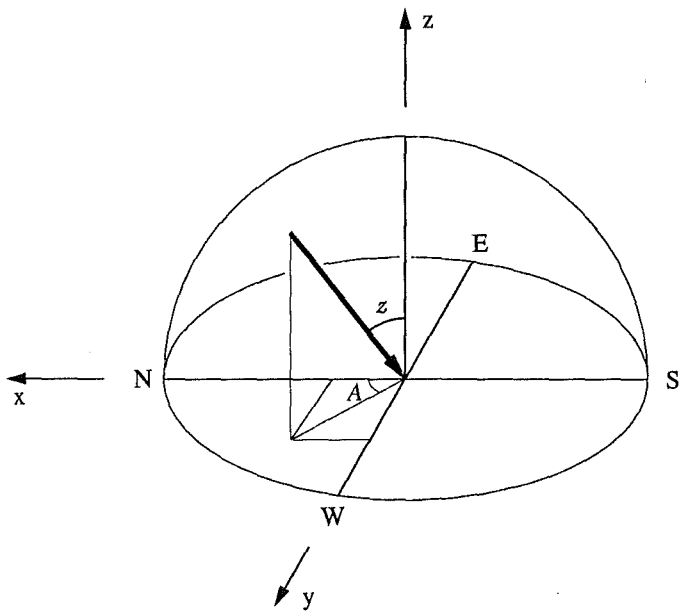


Figure 4.38: The direction vector for the meteor velocity in terms of the azimuth and zenith.

locity of the meteor is

$$\hat{\mathbf{n}} = - \begin{pmatrix} \sin z \cos A \\ \sin z \sin A \\ \cos z \end{pmatrix} \tag{4.16}$$

with the zenith angle, z and azimuth, A as defined in Figure 4.38. To get a radar echo back from a long thin reflector the reflector must be perpendicular to the beam. Mathematically for a position on the ionisation trail, $\mathbf{p}$ this constraint is

$$\mathbf{p}.\hat{\mathbf{n}} \simeq 0 \tag{4.17}$$

This locates the plane in which the meteor is moving. Measuring the time lags between the echo profiles detected at the three different receiver stations will determine the velocity of the meteor projected into the ground plane. The three components of the meteor velocity can then be calculated.

The position vector of the meteor as it forms the ionisation trail may be described as

$$\mathbf{p} = \mathbf{p}_1 + \hat{\mathbf{n}} V_1 t, \qquad t = 0 \quad \text{at} \quad \mathbf{p}_1 \tag{4.18}$$

where V_1 is the magnitude of the observed velocity and t the time the meteor was at that point. The velocity of the meteor is assumed constant as the meteor moves from P_3 to P_1.

The position vectors of the three receiver stations are:
Home site,

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.19}$$

Nutt site,

$$\mathbf{x}_2 = \begin{pmatrix} D_{12}\cos\alpha \\ D_{12}\sin\alpha \\ 0 \end{pmatrix} \tag{4.20}$$

Spit site,

$$\mathbf{x}_3 = \begin{pmatrix} D_{13}\cos\beta \\ D_{13}\sin\beta \\ 0 \end{pmatrix} \tag{4.21}$$

See the overlay on the map in Figure 4.2 for a definition of the symbols used by these position vectors.

The centre of the region on the ionisation trail which reflects the radar beam back to the receiver will be different for each of the three sites. These points are named P_1, P_2 and P_3 for the Home, Nutt and Spit sites respectively and are marked on Figure 4.37. The points P_i are a unique function of the parameter t. We define the time $t = t_i$ when $\mathbf{p}$ of equation (4.18) is the position vector of the point P_i.

The reflection points P_i will occur where the distance from the transmitter (TX) to the trail and back to the receiver (RX) is a minimum. Equation (4.18) is based on the shortest distance, $\mathbf{p}_1$ from the Home site to the meteor path. By definition the point P_1 is at $t = t_1 = 0$. For the other two points, P_2 and P_3, it is more convenient to calculate this minimum from a sum of the squares of the two distances involved.

$$\Sigma(\text{distance})^2 = (\text{distance TX} \longrightarrow P_i)^2 + (\text{distance } P_i \longrightarrow \text{RX})^2 \tag{4.22}$$

Since only the relative time lags between the stations are important it is sufficiently accurate to assume that the transmitter is located at the Home site, $\mathbf{x}_1$. The square of the distance from the transmitter to a general point P_i on the meteor path is

$$\mathbf{p}_i.\mathbf{p}_i = \mathbf{p}_1.\mathbf{p}_1 + V_1^2 t^2 \tag{4.23}$$

since $\mathbf{p}_1$ is normal to $\hat{\mathbf{n}}$ by equation (4.17). The square of the distance from a site i to a general point P_i on the meteor path is

$$\begin{aligned} (\mathbf{p}_j - \mathbf{x}_i).(\mathbf{p}_j - \mathbf{x}_i) &= \mathbf{p}_j.\mathbf{p}_j - 2\mathbf{p}_j.\mathbf{x}_i + \mathbf{x}_i.\mathbf{x}_i & (4.24) \\ &= \mathbf{p}_1.\mathbf{p}_1 + V_1^2 t^2 - 2\mathbf{p}_1.\mathbf{x}_i - 2\hat{\mathbf{n}}.\mathbf{x}_i V_1 t + \mathbf{x}_i.\mathbf{x}_i & (4.25) \end{aligned}$$

Dropping terms constant in t the sum of the distances squared are:
Nutt site,

$$\Sigma(\text{distance})^2_{\text{Nutt}} = \mathbf{p}_2.\mathbf{p}_2 + (\mathbf{p}_2 - \mathbf{x}_2).(\mathbf{p}_2 - \mathbf{x}_2) \quad (4.26)$$
$$\rightarrow 2V_1^2t^2 - 2\hat{\mathbf{n}}.\mathbf{x}_2V_1t \quad (4.27)$$

Spit site,

$$\Sigma(\text{distance})^2_{\text{Spit}} = \mathbf{p}_3.\mathbf{p}_3 + (\mathbf{p}_3 - \mathbf{x}_3).(\mathbf{p}_3 - \mathbf{x}_3) \quad (4.28)$$
$$\rightarrow 2V_1^2t^2 - 2\hat{\mathbf{n}}.\mathbf{x}_3V_1t \quad (4.29)$$

The minimum path from the transmitter to each of the sites will occur when

$$\frac{d}{dt}\sum((\text{distance})_{\text{site}})^2 = 0 \quad (4.30)$$
$$t_1 = 0$$
$$t_2 = \frac{1}{2V_1}\hat{\mathbf{n}}.\mathbf{x}_2 \quad (4.31)$$
$$t_3 = \frac{1}{2V_1}\hat{\mathbf{n}}.\mathbf{x}_3$$

Defining the time lags as $\text{Lag}_{1i} = t_1 - t_i$, the time lags between the Home site and each of the remote sites are

$$\text{Lag}_{12} = t_1 - t_2 = \frac{1}{2V_1}D_{12}\sin z[\cos\alpha\cos A + \sin\alpha\sin A] \quad (4.32)$$
$$\text{Lag}_{13} = t_1 - t_3 = \frac{1}{2V_1}D_{13}\sin z[\cos\beta\cos A + \sin\beta\sin A] \quad (4.33)$$

Defining two variables X_2 and X_3 as

$$X_2 = \frac{\text{Lag}_{12}}{D_{12}}, \qquad X_3 = \frac{\text{Lag}_{13}}{D_{13}}$$

and rearranging gives

$$\tan A = \frac{X_3\cos\alpha - X_2\cos\beta}{-X_3\sin\alpha + X_2\sin\beta} \quad (4.34)$$

Since most of the radiated power from the transmitter antenna is directed south, the azimuth will be in the range $-\frac{\pi}{2} < A < \frac{\pi}{2}$. If the meteor trail occurs to the north of the radar the time lag Lag23 will be positive placing the azimuth in the range $\frac{\pi}{2} < A < \frac{3\pi}{2}$.

From the condition in equation (4.17)

$$R\cos\Psi\sin z\cos A - R\sin\Psi\cos z = 0 \quad (4.35)$$

This gives the zenith angle as

$$\tan z = \frac{\sin\Psi}{\cos\Psi\cos A} \quad (4.36)$$

The zenith angle will always be in the range $0 < z < \frac{\pi}{2}$.

The velocity of the meteor can be calculated from either equation (4.32) or (4.33). In equation (4.37) if X_2 is close to zero then $\cos(\alpha - A)$ is also approximately zero rendering the expression numerically unstable. Equation (4.38) is similarly unstable for small values of X_3. The best numerical answer will be from the expression with the largest value of X_i.
When $X_2 > X_3$,

$$V_1 = \frac{1}{2} \sin z \frac{\cos(\alpha - A)}{X_2} \tag{4.37}$$

and for $X_3 > X_2$,

$$V_1 = \frac{1}{2} \sin z \frac{\cos(A - \beta)}{X_3} \tag{4.38}$$

$$\mathbf{v}_1 = -V_1 \begin{pmatrix} \sin z \cos A \\ \sin z \sin A \\ \cos z \end{pmatrix} \tag{4.39}$$

The meteor velocity, $\mathbf{v}_1$ is now completely defined in the local coordinate frame. Its radiant in terms of the azimuth, equation (4.34) and zenith equation (4.36) and its speed, equation (4.37) or (4.38) are known. The location, orientation and relative velocity of the local coordinate frame with respect to a heliocentric system can be established. A series of coordinate transformations and velocity corrections will give the heliocentric velocity and position of the original meteoroid (see Chapter 6).

4.6.2 Altitude of Reflection Points

The difference between the position of the meteor ionisation trail and the location of the local coordinate frame is negligible in this heliocentric system. The 200 to 400 km displacement is small compared with the Earth's orbital radius. It is neglected as small in the calculation of the meteor orbit, however the actual height of the ionisation trail is of interest for studies of meteor aeronomy. The position vectors of the different reflection points on the trail can be calculated from equation (4.18). The altitude of these points can then be calculated geometrically.

The ionisation height of the Home site reflection point is calculated directly from the elevation angle and range. The difference in height between this point and the remote site reflection points can be found from the vertical component of the meteor velocity vector, v_z.

$$h_1 = \frac{2R_{\text{Home}} R_{\oplus} \sin \Psi + (R_{\text{Home}})^2}{2R_{\oplus} + R_{\text{Home}} \sin \Psi} \tag{4.40}$$

$$h_2 = h_1 - v_z \frac{\mathrm{Lag}_{12}}{\mathrm{Prf}} \tag{4.41}$$

$$h_3 = h_1 - v_z \frac{\mathrm{Lag}_{13}}{\mathrm{Prf}} \tag{4.42}$$

Chapter 5

Fresnel Velocity Measurements

5.1 Introduction and Theory

The presence of a Fresnel diffraction pattern associated with meteor trails is a well established feature of radar meteor detection. It has been extensively used to provide velocity measurements for meteor orbit radars. Ellyett and Davies (1948) first described how to measure meteor velocities using the diffraction of radio waves from trails during formation. This work was continued at Canterbury by Bennett (1953) in a MSc thesis on the measurement of meteoric velocities by the radio method. The Havana radar system used Fresnel patterns to determine meteor velocities for each of their eight receiver sites allowing measurements of atmospheric deceleration to be made. The time difference between patterns from different receiver sites was used to determine the spacing of the specular reflection points (Cook *et al*, 1972).

Using the Fresnel technique for measuring velocities provides an independent check on the time-lag based velocities that form the basis of the AMOR system. This implementation of the technique takes advantage of the digitised amplitude profile to make detailed least squares fits of the Fresnel diffraction pattern. Combining the data from several spaced receivers with velocities determined over subsets of the diffraction pattern allows a measurement of the meteor's atmospheric deceleration to be made. This is a new extension to the established Fresnel method of determining meteor velocities.

The combination of transmitter, meteor trail and receiver is analogous with that of optical diffraction around a straight edge screen. The transmitter acts as a monochromatic point source. Both this source and the viewing point remain fixed as the illuminated section grows longer. The meteor produces the ionisation trail and its motion causes this illuminated trail to grow longer. The change over time in the Fresnel diffraction can be used to measure the velocity of the meteor.

The receivers used by the Christchurch meteor orbit radar are not phase locked to the transmitter. This means that only the amplitude profile is available for Fresnel velocity calculations. As the meteor approaches its point of closest approach, the specular reflection point, the amplitude of the reflected pulse increases. The echo amplitude reaches a maximum shortly afterwards then begins to oscillate. As the meteor moves through each successive Fresnel zone the amplitude contributions reflected by the ionisation trail first add then cancel producing this oscillatory behaviour.

McKinley (1961) gives a mathematical description of the time variation of the echo-power level. (p191, eqns 8.11,8.12).

$$P_r(-s) = \frac{F_r^2}{\pi^2 x^2} \tag{5.1}$$

$$P_r(+s) = 2F_r^2 + \frac{F_r^2}{\pi^2 x^2} + \frac{2\sqrt{2}}{\pi x} F_r^2 \sin\left(\frac{\pi x^2}{2} - \frac{\pi}{4}\right) \tag{5.2}$$

where s is the distance down the trail and, x, the argument of the Fresnel integral. The final echo-power level is $2F_r^2$.

This description uses a greatly simplified model of the meteor echo trail. It assumes the trail is a long thin reflector that extends to $s = -\infty$. The reflector is a stationary column of free elections. The ionisation is sufficiently diffuse so that it may be treated as an underdense trail. In addition the initial radius of the column is small compared with the wavelength and the ionisation level remains constant both along the trail and in time.

The description does give a simple relationship that can be used to study the Fresnel oscillations. After the meteor has moved through the specular reflection point the echo amplitude oscillates about a more slowly varying base level. This base level will decay in the underdense case due to diffusion of the ionisation in the trail. Wind shear and recombination will also affect the shape of the base curve. From equation (5.1) and (5.2) the term $\pi x^2/2$ can be replaced by $2\pi V^2 (t - t_o)^2 / R_o \lambda$. Defining $y(t)$ as the normalised amplitude of oscillation about the more slowly varying mean echo level and a constant, a the oscillations of equation (5.2) are

$$y = \sin\left(a(t - t_o)^2 - \frac{\pi}{4}\right) \tag{5.3}$$

$$a = \frac{2\pi V^2}{R_o \lambda} \tag{5.4}$$

Applying a linear least squares fit to equation (5.3) forms the basis of the method described in this chapter. The final velocity depends on the calculation of the range to the meteor trail, the expression being rearranged to give

$$\left(\sin^{-1} y + \frac{\pi}{4}\right)^{\frac{1}{2}} = \sqrt{a}t - \sqrt{a}t_o \tag{5.5}$$

The quantity of data from the radar system makes it necessary to automate the Fresnel based velocity determinations. Most amplitude profiles show little or no sign of Fresnel oscillations. The code therefore needs to reliably identify suitable echoes as well as calculate Fresnel velocities.

Pecina (1988) provides a more comprehensive scheme using Fresnel characteristics while allowing for atmospheric deceleration. The analysis would provide the pre-atmosphere velocity, v_∞, and the ablation parameter, σ, for the particle directly. The scheme was considered too complex to implement for a calculation that is providing an independent check of the timelag based velocity determinations. The number of observations with good quality Fresnel patterns from all three receiver sites is small, about 1%. The simpler scheme described here can be used to identify high quality Fresnel pattern echoes for future more comprehensive analysis.

5.2 Signal Processing

The first requirement is to establish a mean amplitude level. This needs to work both for echo profiles that exhibit strong Fresnel oscillations and those with none. The algorithm needs to continue establishing the mean level as the frequency of oscillations increases through the pattern. It must cope with noise and not be too distracted by noise spikes.

Calculating the size of each half cycle allows a relatively quick check to determine whether Fresnel oscillations are present. Each successive oscillation should be smaller than the one before. Where the maximum amplitude of an oscillation drops below the noise level the automated fit is abandoned. Provided at least three complete cycles are detected then a useful Fresnel pattern is said to exist and the rest of the reduction attempted.

Previous methods have used the maximum, crossing and/or minimum points to characterise the Fresnel velocity of the meteor. I have modeled each half cycle as a half sine wave and used this to fit intermediate amplitude values. The differences in echo amplitude about the mean level are normalised by dividing by the maximum in each half cycle. By looking at the oscillations about the mean level and then normalising it is possible to allow for non-Fresnel amplitude variations in the profile. No assumptions need be made about the shape of the echo profiles' decay. This approach does implicitly neglect any influence the decay has on the period of oscillation during the pattern. For example an exponential decay does affect the period of the first few oscillations in a minor way, McKinley (1961).

Finally a least squares fit of the data is done and the velocity calculated from equation (5.4). The fit also gives the time t_o that the meteor passed through the specular reflection point. No further use is made of the Fresnel estimate of the time the meteor went through the specular reflection point in

this thesis[1].

To ensure that the velocity fit remains stable over the duration of the Fresnel pattern this same least squares fit is evaluated on subsections of the pattern. If the oscillations of the diffraction pattern become more rapid than the radar pulse rate then the subdomain velocity tends to drop very rapidly. The condition is known as aliasing since the sample rate causes cycles to be missed whilst giving the impression of a longer period oscillation. Checking velocity subdomains provides a final check against aliasing.

Calculating the velocity over successive subdomains of the Fresnel pattern also allows an estimate of the atmospheric deceleration to be made. The amplitude profiles from each of the sites cover different sections of the meteor trail. Where more than one channel gives Fresnel velocities the information can be combined onto a common time base. This gives a series of overlapping velocity determinations from which a linear deceleration can be measured.

The following sections describe each of the steps in detail. Since the Fresnel approach employed in this work has been largely driven by the need to write computer code to evaluate the velocities, the discussion includes considerable reference to its software implementation. This source code is contained in the `OrbFres.Pas` unit. A set of diagnostics programs, Fres_***.Pas, are available to demonstrate the operation of the routines in this unit. An observation with pronounced Fresnel oscillations on all three receiver channels has been used to illustrate the procedures adopted. The details of the fitting routines are displayed using the Nutt site echo profile.

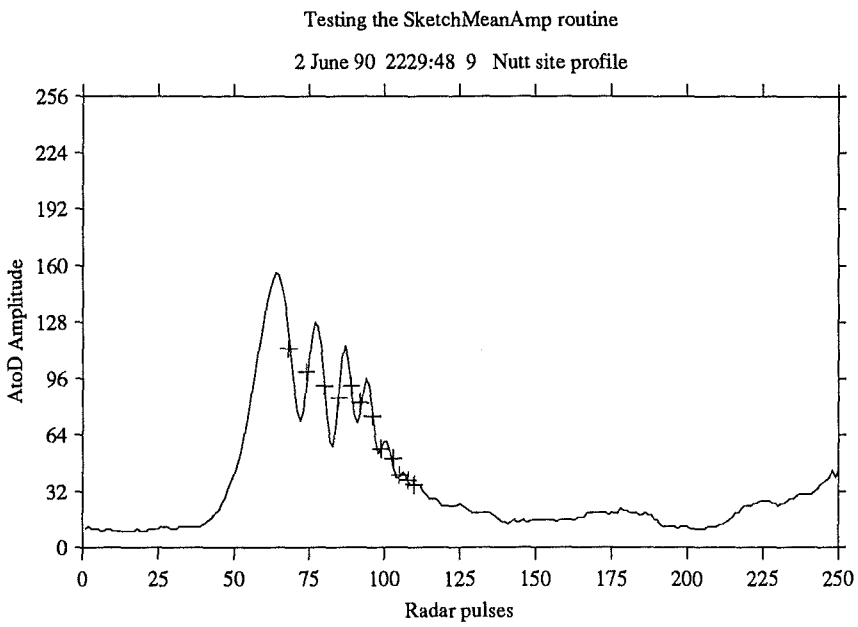
5.2.1 Mean Amplitude Profile

Fitting a mean amplitude profile to the actual echo profile has been split into two sections. In sections dominated by random noise and those where Fresnel oscillations are rapid (cycles of a few radar pulses) a triangular smoothing function did a good job of fitting the mean level. See Section 4.2.2 for the details of the weighted triangular smoothing function employed. Earlier in the pattern where the cycles are longer it is necessary to pick the mid points between successive maxima and minima and use these as the basis of the mean amplitude level. The general scheme can be seen in Figure 5.1. The procedure SketchMeanAmp calculates the mean amplitude profile.

The data reduction associated with this project uses despiked amplitude profiles. Noise spikes are identified by comparing the echo amplitude of each individual pulse with its immediate neighbours. Where it is substantially different from the average of the two neighbours a noise spike is said to exist. Occasionally this despiking process removes rapid high amplitude Fresnel oscillations. Oscillations this rapid are getting close to the point where aliasing

[1] Forgotten about it at the time. It might provide an interesting more direct check on the rising edge time lag measurements.

(a)



(b)

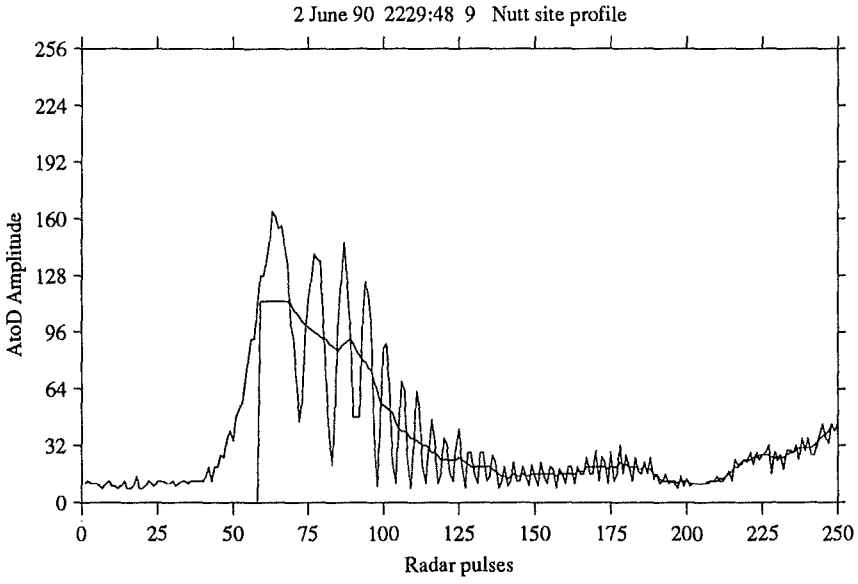


Figure 5.1: Intermediate stages as the SketchMeanAmp routine establishes a mean amplitude profile. (a) Profile smoothed with a weighted triangular smoothing function (R = 3). The plus symbols mark the mid points between successive maxima and minima. (b) The rough mean amplitude profile has been overlaid on the raw echo. The final smoothed form can be seen on Figure 5.2.

will occur. That is, the period of oscillation approaches the sampling rate. The despiked profiles help reduce error from aliasing.

The routine starting at the initial maximum looks for each successive local maxima and minima. The mid points between each of these pairs are used as estimates for the crossing points of the oscillations. To reduce the effect of noise fluctuations this is done using a smoothed profile (R=3). Each of these crossing points should get successively closer together. Where this stops happening the program switches back to the smoothed profile. Unless this is done the 'mean' amplitude level tends to lose track of base level and tries to follow larger scale fluctuations. While the crossing points can be identified the routine simply joins them with a straight line. Since each of the half cycles is fitted with a half sine curve this is probably a sufficiently good fit of the mean amplitude levels. The profile is extended back from the first crossing point at a constant amplitude. This is done to ensure the profile starts before the area of interest is reached. The Fresnel velocity determination remains sensitive to the computer based selection of the crossing points. However the technique does allow considerably more information to be used for the final fit.

The mean amplitude profile at this stage looked a bit rough. For aesthetic reasons rather than anything else I added an additional smoothing to produce the final profile. The mean amplitude profile reflects the echo amplitudes in the absence of the Fresnel diffraction effect. To provide a basis for further calculation the mean amplitude is subtracted from the raw echo amplitude. This is called the residual amplitude profile.

5.2.2 Amplitude Oscillations

Profiles where Fresnel oscillations exist need to be identified. In all cases oscillations are assumed to start at the location of the profile maximum. This usually corresponds to the first maximum of the pattern. Occasionally the second peak is larger and there is an earlier peak. A crude search back to find previous peaks is made. When the code misses the first maximum and the search back is not successful the final result is mostly rejected because the velocities calculated over the subdomains indicate an unstable result. Missing the first maximum leaves the phase angle 2π out throughout the pattern.

Starting from this maximum the size of each of the half cycles is determined by the routine SizeHalfCycles. The routine FindEndCycle uses this information to help identify the effective end of the useful pattern. The Fresnel diffraction pattern exhibits a steadily increasing rate of oscillation. Physically each successive Fresnel zone becomes shorter as the meteor moves further from the specular reflection point. The recorded amplitude profile should reflect this. Where this criteria breaks down the routine decides it has found the end of the Fresnel pattern. To ensure the routine is relatively insensitive to noise fluctuations and to poorly determined crossing points it actually compares the size of two half cycles with that of the previous two. When

`(ThisTwo < LastTwo + 2)` turns false the pattern is ended. The program also terminates the pattern once a maximum of 60 half cycles is reached. With the radar pulse rate (379 Hz), typical meteor velocities and a half second maximum of recorded data in the profile the 60 half cycles are unlikely to be a restriction.

Where the size of oscillations drop to less than plus or minus four A/D amplitudes the pattern is assumed to have fallen into the noise. In fact, viewing the data by eye, oscillations can often be discerned after this point. However the increased computational uncertainty of including the extra points makes any gains marginal.

The combination of defining the end of the pattern where the period of the oscillation increases unexpectedly or its amplitude gets too small does a good job of including only reliable sections of the Fresnel diffraction pattern. It also seems reliable in placing the end of the pattern before aliasing effects start to appear. If the data record has missed a half cycle because the sampling rate is too slow then all the subsequent phase angle determinations will be one cycle out.

To ensure that chance combinations of profile variations and noise spikes do not cause erroneous velocity determinations, the program requires a minimum size to the pattern. Where less than six half cycles are detected the profile is rejected as having no useful Fresnel diffraction pattern. A small fraction of profiles still erroneously identify Fresnel diffraction patterns at this level. Rejecting all profiles with less that ten half cycles would eliminate all the bad examples I have seen and provide a high quality data set of Fresnel velocities.

The useful end of the Fresnel diffraction pattern is established by the routine FindEndCycle. Figure 5.2(a) displays the output from this routine for the example profile. The amplitude of the cycle following the end of pattern at radar pulse 139 drops below the four A/D amplitude threshold. The attendant ticks at the top of the display mark each pulse of data and indicate the output of the routine SizeHalfCycles.

The oscillating amplitude is being modelled as a sine function with a period steadily decreasing in time. Each point on the oscillating amplitude profile can be uniquely defined by a phase angle. Mathematically this phase angle is $(\sin^{-1} y + \frac{\pi}{4})$ from equation (5.5). The relative phase angle of the oscillation at each radar pulse needs to be determined.

Within the simple theory used, the meteor passes the specular reflection point when the echo amplitude reaches a point halfway up the rising edge to the first maximum. This is $\frac{\pi}{4}$ before the point where the phase angle describing the oscillations is zero. At the first maximum the phase angle has the value $\frac{\pi}{2}$. At the first crossing point where the echo amplitude drops below the mean level this is π and at the first minimum $\frac{3}{2}\pi$. Across each complete cycle thereafter it increases by 2π. Each half cycle is fitted by a half sine curve of constant period. The phase angle at each radar pulse can then be interpolated.

Looking at the meteor profiles gathered by the radar system it is not always

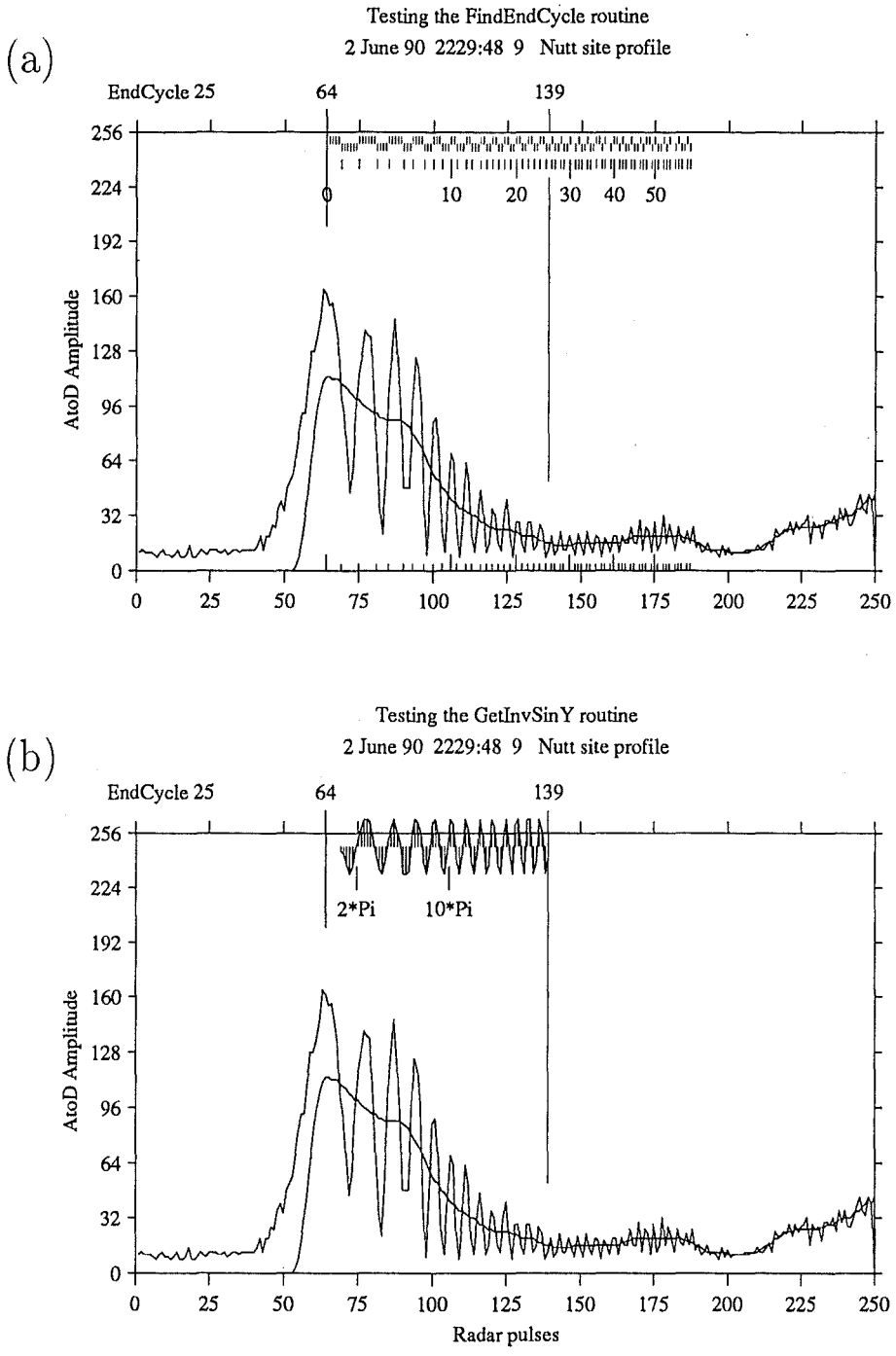


Figure 5.2: Diagnostic output for routines establishing the relative phase angle of the amplitude from each radar pulse within the usable Fresnel diffraction pattern. (a) After processing by FindEndCycle showing information from SizeHalfCycles and the numbering for each half cycle. (b) Test display of a sine function based on the phase angles calculated in the GetInvSinY routine.

clear where the point corresponding to the zero phase angle is located. Even more difficult is deciding a mean amplitude level (in the absence of Fresnel effects) for the initial maximum. Various assumptions suggest that the theory does not model very well this region around the specular reflection point and the first Fresnel zone. No attempt is made to include data from the first half cycle in the velocity fit.

5.3 Atmospheric Deceleration

A least squares fit based on equation (5.5) is used to calculate the Fresnel velocity. The fit uses data from the first crossing point (phase of π) to the end of the pattern. To check that the calculated velocity remains stable over this range the velocity in selected subdomains is evaluated.

As the meteor moves through the atmosphere it is subject to a deceleration. Each successive oscillation in the diffraction pattern is due to the meteor's motion through the corresponding Fresnel zone. The meteor's velocity as it moves through each successive Fresnel zone will therefore be reduced. Comparing the implied velocity at the beginning of the pattern with that at the end allows an estimate of the deceleration to be made.

The geometric layout of the three receiver stations means that the specular reflection points associated with each site are at different locations along the meteor's flight path. The precise regions over which the Fresnel velocities are calculated are therefore different for each site. Combining the data from profile subdomains of different sites gives an overlapping set of velocities for the meteor at different points along its path.

Specifically each diffraction pattern is broken up into overlapping subdomains centered on the pulse closest to each crossing point. They extend for one complete cycle on either side of these points. The velocity subdomain calculations and stability checks are done within the procedure GetFresnelDecel. Where the velocity drops by 5 km.s^{-1} from one point to the next aliasing is assumed and the end point of the pattern moved accordingly. An estimate of the deceleration for the individual profile is also made. Where this is greater than 120 km.s^{-2} the Fresnel velocity fit on that profile is abandoned. These decelerations are not always very reliable estimates of the atmospheric deceleration, especially for short patterns. Figure 5.3 gives a graphical display of the subdomain velocities associated with the example profile.

Each amplitude profile contains 250 echo amplitudes. The observation code places the meteor detection point for that profile at index location 50. To ensure that the data from different stations are combined correctly, profiles are converted to the time base of the station with the earliest detection. Where at least two profiles exhibit Fresnel diffraction effects an estimate of the atmospheric deceleration is made. The code requires a total of at least eight subdomain velocity values before proceeding with the calculation. Figure 5.4

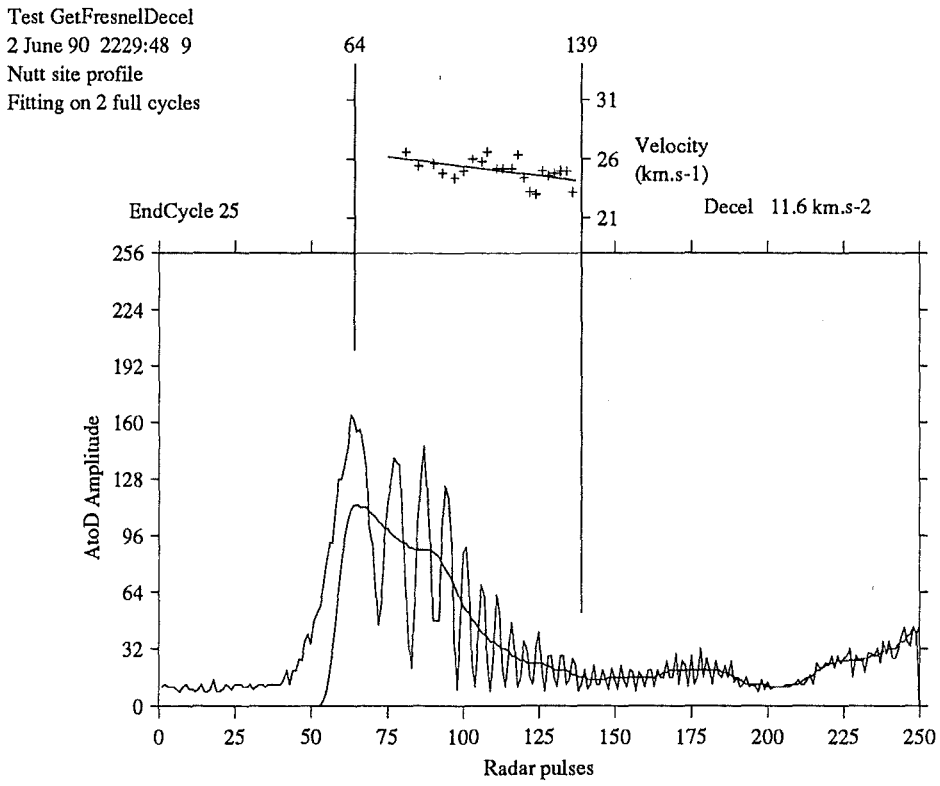


Figure 5.3: Velocity over two cycle subdomains. The scale is centered on the Fresnel velocity for the whole profile. The best fit deceleration gives 12.3 km.s^{-2} measured over 0.15s.

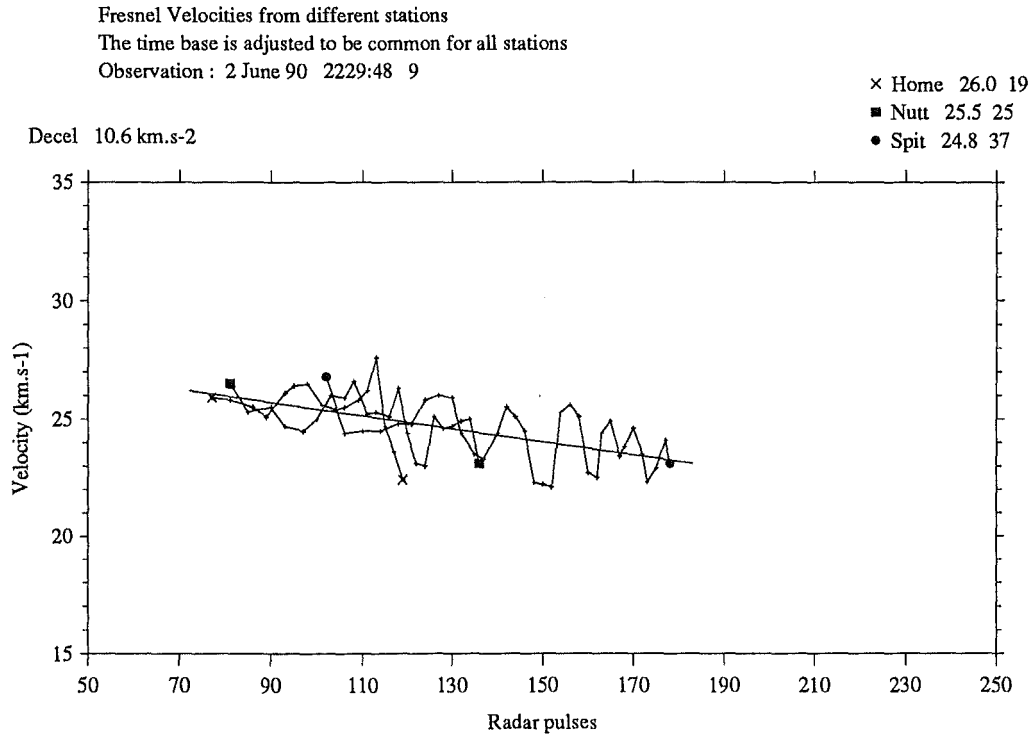


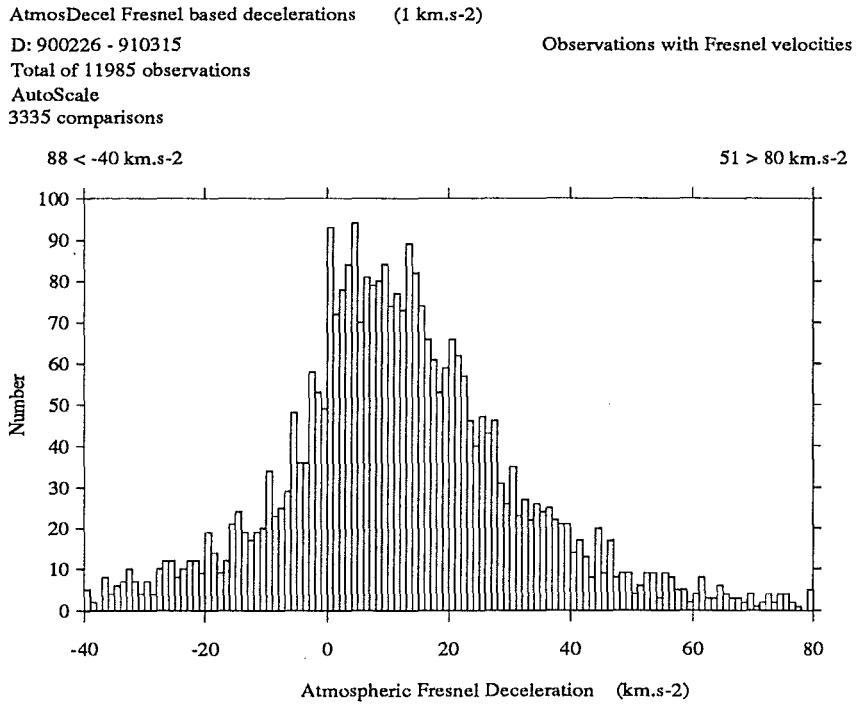
Figure 5.4: Velocity data combined from the simultaneous observation of Fresnel diffraction patterns at three receiver sites. An atmospheric deceleration of 10.7 km.s^{-2} measured over 0.26s. The timelag based velocity for this observation was 28.2 km.s^{-1}.

combines the data from all three sites for the example observation. The deceleration remains constant throughout the duration of the Fresnel pattern. The spread of the individual subdomain velocities is approximately 3 km.s^{-1}.

Figure 5.5(a) plots the distribution of atmospheric decelerations. Most of the atmospheric decelerations measured are in the range from -5 to 40 km.s^{-2}. The negative value probably reflects an accuracy limit for the method. There are some fairly convincing observations of meteors with decelerations of around 100 km.s^{-2}, see Figure 5.6. A check that the Fresnel velocity on each of the profiles are similar is make to ensure that deceleration of this order are not the spurious result of two profiles with completely different decelerations. The configuration of the radar system selects strongly against detecting meteors with Fresnel patterns showing large atmospheric decelerations. With high rates of deceleration the meteor will tend to burn out very rapidly. It is therefore unlikely that enough oscillations will be recorded to allow a Fresnel velocity to be determined, let alone a deceleration.

The atmospheric decelerations in Figure 5.5(a) can be compared with a

(a)



(b)

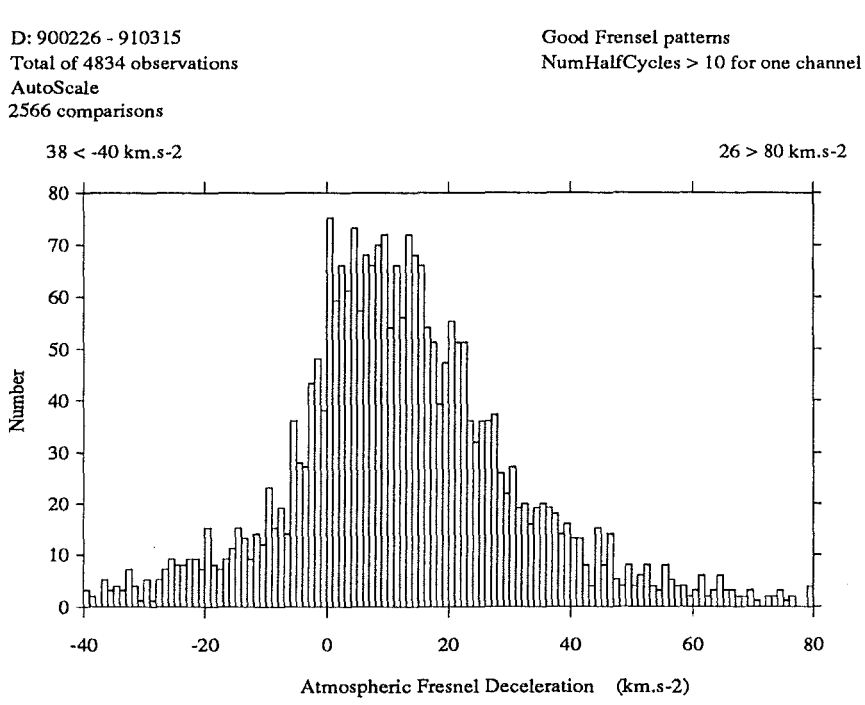


Figure 5.5: Atmospheric deceleration measurements based on the Fresnel diffraction patterns. Many of the more extreme values (especially negative decelerations) are due to calculation errors. However some very high decelerations do seem from the data to be realistic. The lower panel plots a high quality data set where one profile must be at least 10 half cycles long and the multiple Fresnel speeds agree to within 5 km.s^{-1}.

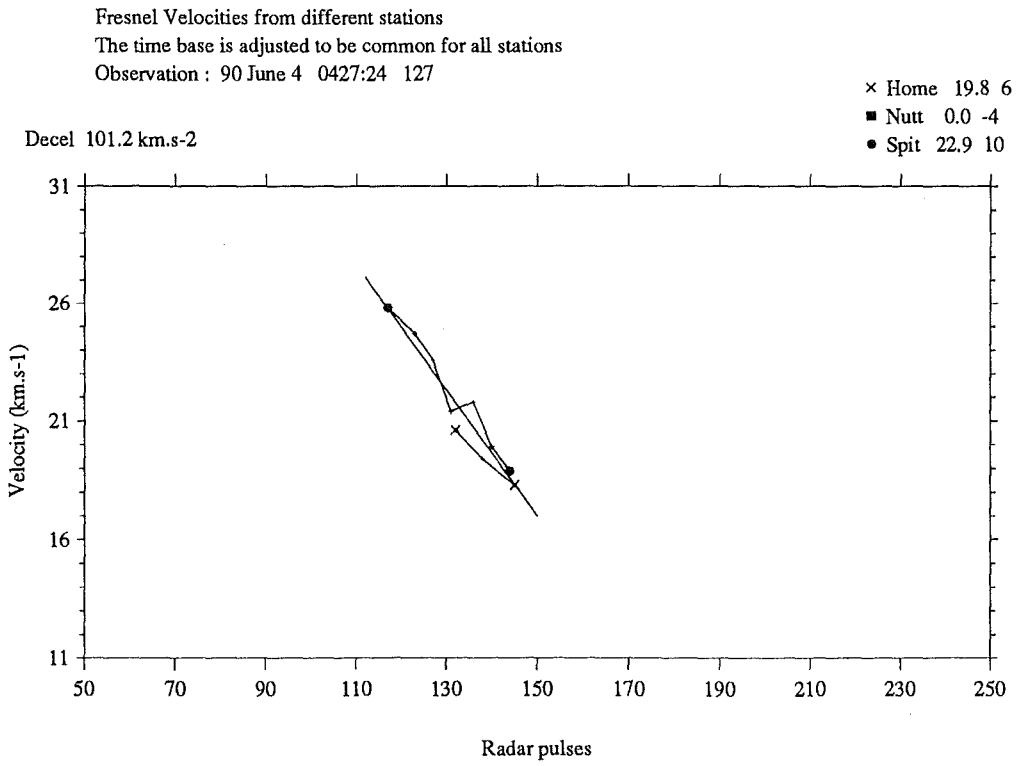


Figure 5.6: The meteor observation shows a very rapid atmospheric deceleration of 101.2 km.s^{-2}.

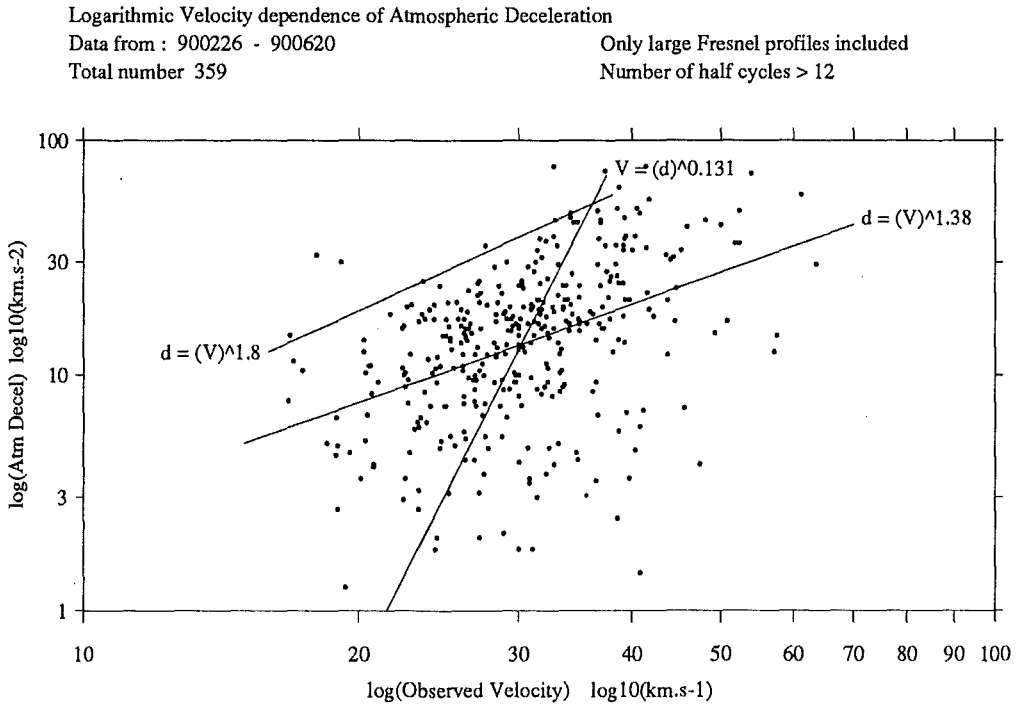


Figure 5.7: The atmospheric deceleration plotted as a function of the observed velocity. The two regression lines cross at the mean point (30 km.s^{-1},14 km.s^{-2}). The third curve fits the low velocity, high deceleration cutoff.

high quality subset plotted in the bottom panel. Only observations where one profile is at least 10 half cycles long and the multiple Fresnel speeds agree to within 5 km.s^{-1} are included in this high quality subset.

5.4 Ablation Coefficient

The atmospheric deceleration for a meteor needs to be estimated and allowed for in calculating the original orbit of the meteoroid. A discussion of the theoretical basis behind this correction is found in Section 6.3. An ablation coefficient, σ, can be evaluated using the deceleration data from the Fresnel methods described here. To reduce contamination from spurious results the high quality data set is used.

Verniani (1973) infers from observations a dependence of $\sigma \sim v_\infty^{-2.3}$ for radio meteors, a relation he considers to be too severe due to an initial radius height ceiling effect. In this work he measured physical parameters for radio meteors and faint photographic meteors (Verniani, 1967) finding a distribution

of σ values. The ablation coefficient for the magnitude +8 radio meteors is about 5 times smaller than for the photographic ones. Assuming $\sigma \sim v_\infty^{-2.3}$ the radio measurements were reduced to $\log \sigma = -11.8$ for $v_\infty = 35$ km.s^{-1}. The resultant ablation values are essentially independent of the initial mass.

Bronshten (1983) summarises the velocity and mass dependence of σ from a variety of authors (Fig.13 and 14, page 45). A dependence of $\sigma \sim v_\infty^{-1.6}$ fits the data plotted in his Fig.13. By normalising about 35 km.s^{-1} the ablation coefficient can be expressed in the form

$$\begin{aligned} \sigma &= 2.5 \times 10^{-12} \left(\frac{35}{v}\right)^{1.6} && \left[(cm)^{-2}.s^2\right] \\ &= 0.025 \left(\frac{35}{v}\right)^{1.6} && \left[(km)^{-2}.s^2\right] \end{aligned} \qquad (5.6)$$

Using the AMOR data set the dependence of atmospheric deceleration on observed velocity is plotted in Figure 5.7. The uncertainty in observed velocity will just reflect the measurement uncertainty of the Fresnel method. The scatter in atmospheric deceleration will come from a number of sources. How close the reflection point is to the point of maximum ionisation, the altitude of the meteor, its physical composition, its tendency to fragment as well as the uncertainty in the measuring method will all contribute. This suggests a greater intrinsic variation in the deceleration values. The two regression lines, $d \sim v^{1.38}$ (all variation in deceleration) and $v \sim d^{\,0.131}$ (all variation in velocity) provide limits to the relationship. The third line on Figure 5.7 fits the low velocity high deceleration cutoff of the distribution. This edge will be the least sensitive to the selection effects involved in collecting the data.

The deceleration is inversely proportional to the ablation coefficient; see equation (6.30). A relationship between deceleration and velocity based on Bronshten of $d \sim v^{-1.6}$ is quite consistent with the data from the current survey. This dependence is adopted for the atmospheric deceleration correction necessary in the orbit calculations.

The intersection of the two regression lines in Figure 5.7 is just the mean point at (30 km.s^{-1},14 km.s^{-2}). The mean zenith angle for these observations is around 30°[2]. Combining this information with equation (6.30) and assuming a scale height, H = 7 km gives a value for the ablation coefficient of $\sigma = 0.013 \text{km}^{-2}.\text{s}^2$ at 30 km.s^{-1}. This is slightly lower than previous workers have obtained. Our radar will detect meteors down to magnitude +12. This is fainter than previous studies and the ablation coefficient might be expected to be less. See the comparison of Verniani's data discussed above.

For the data obtained in this survey the ablation coefficient seems to be best expressed as

$$\sigma = 0.013 \left(\frac{30}{v}\right)^{1.60} \qquad (5.7)$$

[2]It should be noted that the zenith angle is based on timelag velocity determinations.

$$= 3.00\,(v)^{-1.60} \qquad \left[(km)^{-2}.s^2\right] \qquad (5.8)$$

The value estimated here comes from a very limited subset of the reduced observations. Only relatively slow meteors give useful Fresnel patterns at the AMOR pulse rate. Calculating decelerations requires reasonably long Fresnel patterns on at least two channels. Particle fragmentation and rapid decelerations are both eliminated from the high quality data set used to estimate the ablation coefficient.

An estimate of the ablation coefficient is necessary to make the correction for atmospheric decleration. The value from Bronshten, equation 5.6 is half the ablation coefficient measured here. Both values seem plausible and the uncertainty between them gives an error of 1 or 2 km.s^{-1} in the meteor velocity before it entered the atmosphere.

5.5 Timelag Verifications

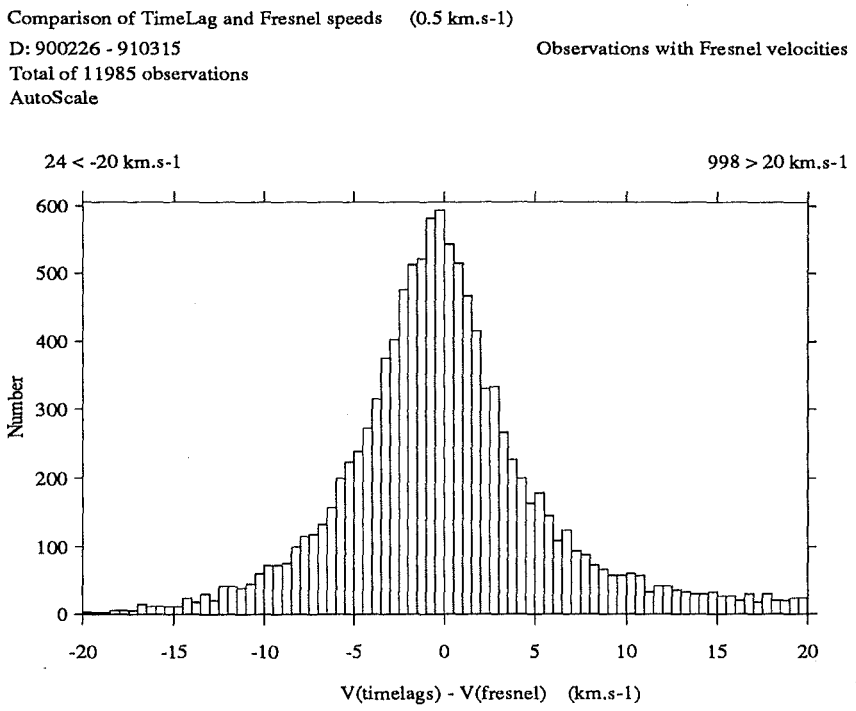
One of the primary reasons for pursuing the Fresnel diffraction method of calculating the meteor velocity was to provide an independent check on the timelag based velocities. A specific example of this comparison can be seen for the observation 1990 June 2, 2229:48 #9, in Figure 5.4. Here the two methods give an agreement to within 2 km.s^{-1}.

A comparison between the timelag and Fresnel speeds, for the complete data set can be seen in Figure 5.8. The differences between the two are reasonably evenly spread around zero. The two techniques give similar answers. The timelag based method yields velocities in many more cases than techniques using Fresnel diffraction patterns. This is largely because Fresnel patterns are absent in such a large fractions of the observations.

Figure 5.8(b) compares velocity data with more substantial Fresnel patterns only. This higher quality data set where at least five full Fresnel oscillations are present allows the velocity to be determined to greater accuracy. It can be seen that the number of wildly dissimilar speeds is reduced. A significant fraction of the larger disagreements in Figure 5.8(a) can be attributed to erroneous Fresnel velocity determinations. The distribution in Figure 5.8(a) appears slightly skewed indicating the calculated Fresnel speeds are higher than those based on the three station timelags. This largely disappears where longer Fresnel diffraction patterns are compared.

In comparing timelag with Fresnel velocities it should be remembered that a small number of Fresnel velocities are just plain wrong (possibly 5% to 10%). Usually the result seems good to within 3 km.s^{-1}. See Figure 5.4 for an example. This estimate of uncertainty was obtained by examining the spread of subdomain velocities for a large number of observations. The distribution in Figure 5.8 is approximately Gaussian with a total width of 7 km.s^{-1} measured at 0.607 of the maximum. The halfwidth at this level is an estimate of one sigma standard deviation for the distribution. For a Gaussian distribution the

(a)



(b)

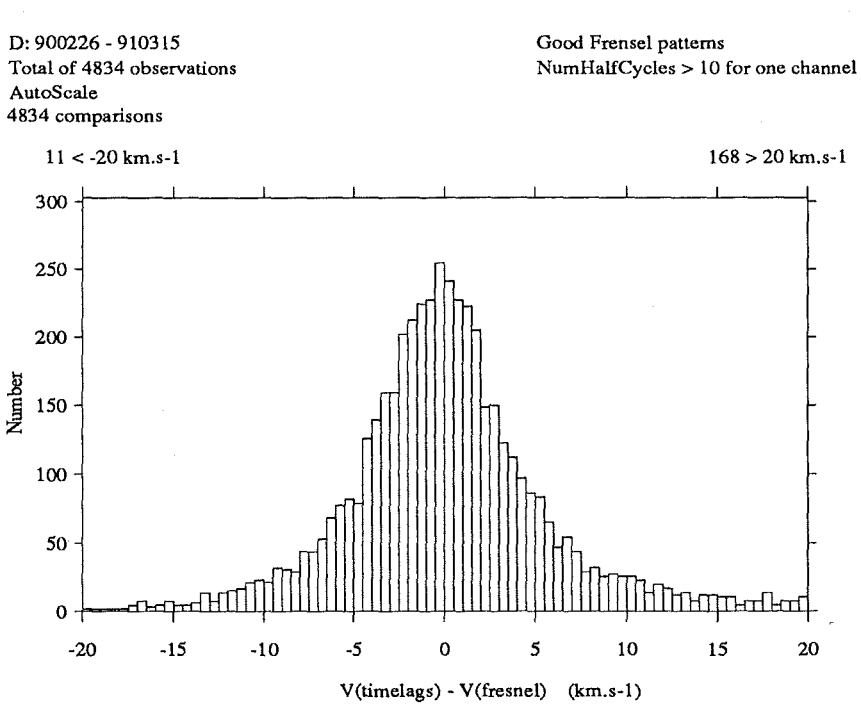


Figure 5.8: Comparison of the Time-Lag and Fresnel speeds. (a) Uses all data for which Fresnel velocities are determined. 11985 comparisons included. (b) Only includes those Fresnel velocities where a profile contains more than 10 half cycles. 4834 comparisons included.

sum of the component standard deviations adds as the square of the individual deviations.

$$\sigma^2_{\text{total}} = \sigma^2_{\text{Fresnel}} + \sigma^2_{\text{timelags}} \quad (5.9)$$

Assuming that the uncertainties from each are similar, the standard deviation for the individual velocity distributions is 2.5 km.s^{-1}. This agrees well with the more qualitative assessment noted above. This suggests an uncertainty of same order, 3 km.s^{-1} can be attributed to the timelag velocity determinations. The 1990 η Aquarid shower was used as a calibration of the AMOR system. The velocity measurement uncertainty estimated here agrees with that of the timelag error for η Aquarid meteors discussed in Section 7.4.1.

5.6 Hyperbolic Meteor Velocities

A reasonably large number of observations exist that indicate that the meteor had a velocity greater than the solar escape velocity at the Earth's radius. High velocity meteors have correspondingly shorter timelags. This increases uncertainties in the velocity determination. Heliocentric velocities within a few km.s^{-1} of the solar escape velocity (42 km.s^{-1}) could be due to the uncertainty in timelag velocities just discussed[3]. Hyperbolic orbits would be erroneously indicated in these cases. To check whether more extreme velocity measurements were valid, observations with heliocentric meteoroid velocities greater than 48 km.s^{-1} were investigated.

The measured timelag velocities were compared with Fresnel based speeds. Figure 5.9 plots the results of this comparison for fast meteors. The 26 observations where the two agree to better than 8 km.s^{-1} are highlighted. 26 out of 11985 observations with Fresnel velocities seem to be hyperbolic. Six of these are hyperbolic and agree with the timelag velocities to better than 3 km.s^{-1}. It must be reiterated that Fresnel methods for amplitude profiles collected with the 379Hz pulse rate of the AMOR system are poor at establishing velocities for fast meteors. None of the hyperbolic group come from high quality Fresnel patterns. I suspect that many of the low velocity Fresnel patterns are the result of slow moving fragments formed as the particle breaks up in the atmosphere. These give small amplitude variations that are in severe disagreement with good quality rising edge timelag velocities. The fraction of hyperbolic meteors detected would be a lower limit for hyperbolic meteors detected. A total of 559 observations had differences of greater than 8 km.s^{-1}, 396 of these were more than 20 km.s^{-1} different. Some of the timelag based hyperbolic meteor velocities are confirmed as real by the Fresnel velocity measurements. The results are heavily contaminated with bad velocity determinations.

[3]The great mystery from *Hitch-Hiker's Guide to the Galaxy* is finally solved. The answer to Life, the Universe and Everything was found to be 42. Some doubt remained about what exactly the question was. I can now answer it. If the Earth were going faster than 42 km.s^{-1} it would leave the solar system behind; FOREVER.

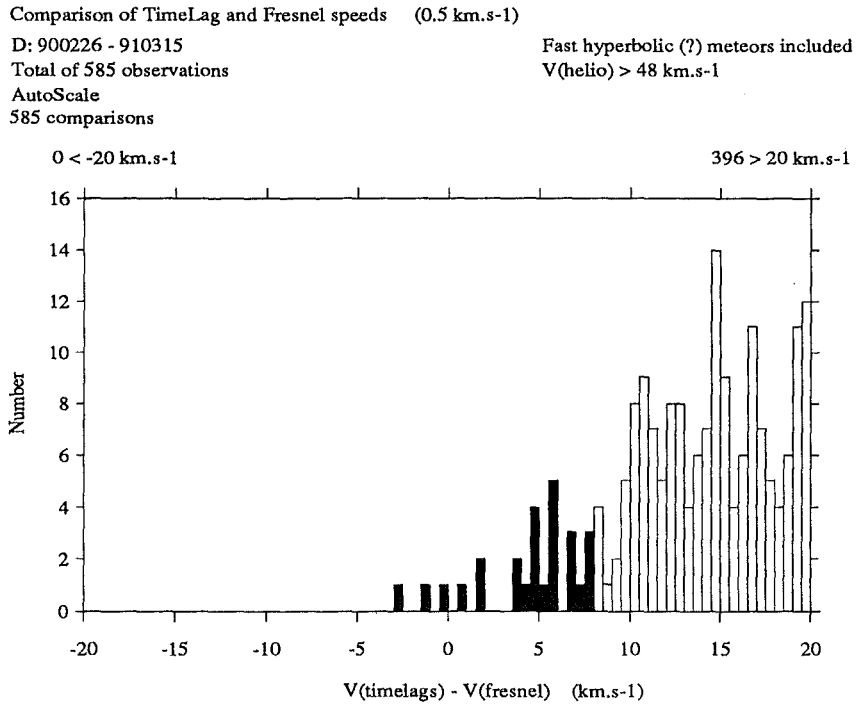


Figure 5.9: Comparison of the timelag and Fresnel speeds for observations with a heliocentric velocity of more than 48 km.s^{-1}. The filled bars indicate where the difference between the two velocities is less than 8 km.s^{-1}. A total of 26 observations are included in this range. It compares with 559 observations where the difference is greater than 8 km.s^{-1}.

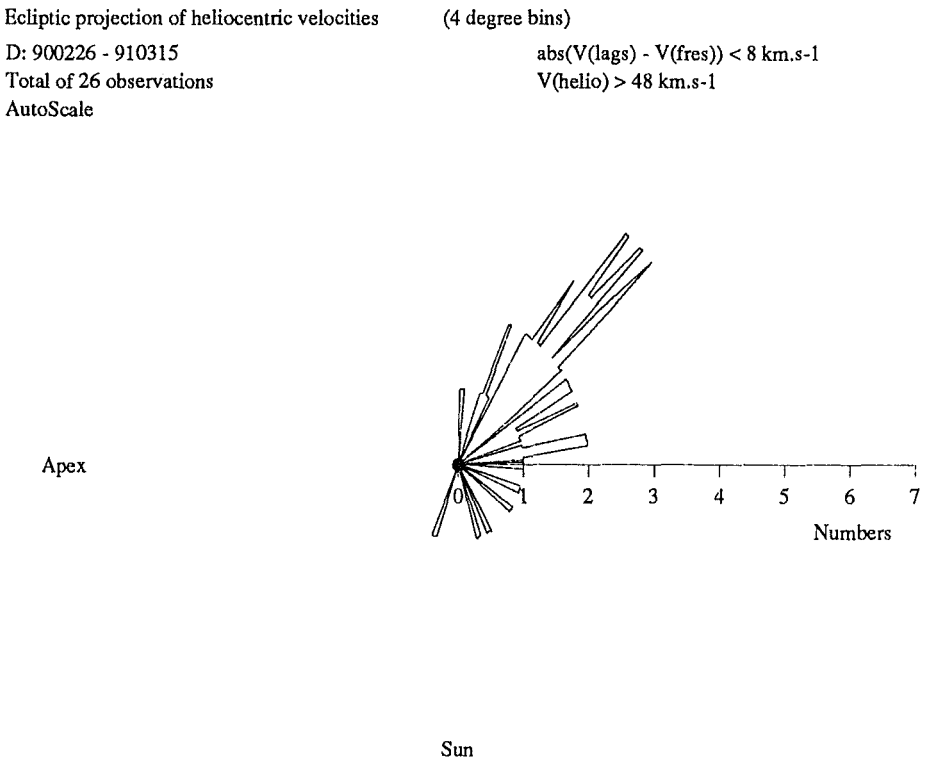


Figure 5.10: The ecliptic projection of heliocentric meteoroid velocities; where they are coming from. The timelag based speed is greater than 48 km.s^{-1}. Points are plotted where the Fresnel and timelag speeds agree to within 8 km.s^{-1}; a total of 26 observations. 559 observations have a difference of greater than 8 km.s^{-1}.

Of the fast meteors with good agreement between Fresnel and timelag velocities most were prograde and near the plane of the ecliptic. This is almost certainly a selection effect. The orientation of the radar beam preferentially detects particles from near the ecliptic. The hyperbolic meteoroids are catching the Earth, reducing the impact velocity, making a Fresnel velocity determination more probable. Figure 5.10 plots the ecliptic projection of the heliocentric velocity of these definitely hyperbolic meteors. Problems with sporadic-E echoes and broadcast interference during the day mean most meteors are detected at night. More particles will be detected approaching perihelion. To assess whether these particles are inter-stellar in origin the distribution of the definitely hyperbolic meteors needs to be sampled around the Earth's orbit. Each part of the orbit will tend to sample possible interstellar particles from a different direction. If the distribution remains random in longitude but with low inclination then a solar system origin is more likely, possibly the result of interparticle collisions.

Chapter 6

Calculation of the Orbital Elements

Five orbital parameters are sufficient to fix a meteoroid's orbit in space. From the radar observations we know the velocity vector of the meteor. The time of observation is also recorded. Several distinct corrections to the velocity are necessary to allow for atmospheric deceleration, the Earth's motion and the Earth's gravitational field. A number of coordinate transformations are desirable to aid in calculating the velocity corrections and facilitate establishing the eventual orbital elements.

Traditionally the calculation of the meteor orbit is done as a series of corrections to the observed radiant position and scalar velocity. These aberrations are well known and have been described by Porter (1952) and again by Lovell (1954). Neither of these authors give a detailed derivation of the correction formulae. A number of smaller velocity corrections are ignored completely, calculated only partially or evaluated in an inappropriate order.

The traditional approach is however well suited to manual calculations. It relies substantially on using trigonometric functions with associated decisions about which quadrant the radiant is in. Simplifications are sought to reduce the calculation effort.

The derivation presented here keeps track of the velocity vector of the meteor. Matrices are defined to rotate the axes to new coordinate frames. Transformations of the velocity vector are then simply calculated by matrix multiplication. Corrections are introduced to remove the local perturbations on the meteor path due to the Earth. The heliocentric position associated with this unperturbed velocity is assumed to be that of the centre of the Earth at the time of observation.

This new approach does more straight multiplication and addition. The quadrants are implicitly contained in the vector definitions. The resulting algorithm is suited to computer based calculation. Section 6.10 gives a worked example of the orbit calculation scheme. Relevant equations of motion are

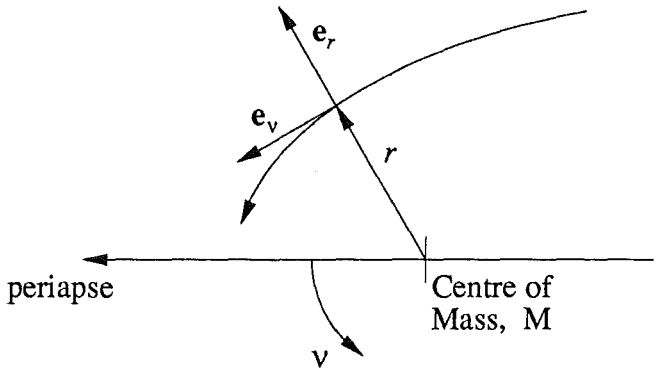


Figure 6.1: Definitions of the polar coordinates used to calculate the equation of motion for the meteoroid. ν is shown positive for an anticlockwise motion.

discussed in the next Section 6.1 before the main argument begins.

6.1 Equations of Motion

The equations of motion in a central gravitational field are presented here. This section is a description of results using the notation of this thesis rather than a strictly formal derivation of them.

The equation of motion may most simply be established in the plane of the orbit. Using polar coordinates centred on the large central mass the radius vector of the orbiting particle is

$$\mathbf{r} = r\mathbf{e}_r \tag{6.1}$$

where $\mathbf{e}_r$ is the unit radius vector. Figure 6.1 shows the location and sense of the polar coordinate frame. The rate of change for the unit vectors $\mathbf{e}_r$ and $\mathbf{e}_\nu$ are

$$\frac{\partial \mathbf{e}_r}{\partial t} = \frac{\partial \nu}{\partial t}\mathbf{e}_\nu \tag{6.2}$$

$$\frac{\partial \mathbf{e}_\nu}{\partial t} = -\frac{\partial \nu}{\partial t}\mathbf{e}_r \tag{6.3}$$

From equation(6.1) the velocity and acceleration of this particle follows as

$$\mathbf{v} = \frac{\partial \mathbf{r}}{\partial t} = \frac{\partial r}{\partial t}\mathbf{e}_r + r\frac{\partial \nu}{\partial t}\mathbf{e}_\nu \tag{6.4}$$

$$\mathbf{a} = \frac{\partial^2 \mathbf{r}}{\partial t^2} = \left(\frac{\partial^2 r}{\partial t^2} - r\left(\frac{\partial \nu}{\partial t}\right)^2\right)\mathbf{e}_r + \left(r\frac{\partial^2 \nu}{\partial t^2} + 2\frac{\partial r}{\partial t}\frac{\partial \nu}{\partial t}\right)\mathbf{e}_\nu \tag{6.5}$$

In the central gravitation field caused by a mass M the equations of motion become

$$\frac{\partial^2 r}{\partial t^2} - r\left(\frac{\partial \nu}{\partial t}\right)^2 = -\frac{GM}{r^2} \tag{6.6}$$

$$r\frac{\partial^2 \nu}{\partial t^2} + 2\frac{\partial r}{\partial t}\frac{\partial \nu}{\partial t} = 0 \tag{6.7}$$

Multiplying equation (6.7) by r, rearranging and then integrating gives

$$\begin{aligned} \frac{\partial}{\partial t}\left(r^2\frac{\partial \nu}{\partial t}\right) &= 0 \\ r^2\frac{\partial \nu}{\partial t} &= h \end{aligned} \tag{6.8}$$

where the constant of integration, h is twice the areal velocity.

The motion of a particle in the gravitational field will be along the locus of a conic section. The general equation for a conic is a good trial solution to equation (6.6). For the semi-latus rectum p, eccentricity e and coordinates as in Figure 6.1 this is

$$r = \frac{p}{1 + e\cos\nu} \tag{6.9}$$

From equation (6.9)

$$\begin{aligned} \frac{\partial r}{\partial t} &= \frac{pe\sin\nu}{(1+e\cos\nu)^2}\frac{\partial \nu}{\partial t} \\ &= \frac{eh\sin\nu}{p} \\ \frac{\partial^2 r}{\partial t^2} &= \frac{eh^2}{r^2 p}\cos\nu \end{aligned} \tag{6.10}$$

Substituting into the equation of motion in a central field, equation(6.6) and using the expression for the areal velocity, equation (6.8) gives

$$\begin{aligned} \frac{GM}{r^2} &= \frac{h^2}{r^2}\left(\frac{1}{r} - \frac{e\cos\nu}{p}\right) \\ &= \frac{h^2}{pr^2} \end{aligned} \tag{6.11}$$

This gives the semi-latus rectum,

$$p = \frac{h^2}{GM} \tag{6.12}$$

The semi-major axis, a, is half the distance between the periapse and the antiapse. It follows from equation (6.9) and Figure 6.1 that

$$\begin{aligned} a &= \tfrac{1}{2}\left(r_{\nu=0} + r_{\nu=\pi}\right) && (6.13)\\ &= \tfrac{1}{2}\left(\frac{p}{1+e} + \frac{p}{1-e}\right) \\ &= \frac{p}{1-e^2} && (6.14) \end{aligned}$$

The scalar velocity at any point on the meteor path now follows from equation (6.4).

$$\begin{aligned} \mathbf{v}.\mathbf{v} = V^2 &= \left(\frac{\partial r}{\partial t}\right)^2 + r^2\left(\frac{\partial \nu}{\partial t}\right)^2 && (6.15)\\ &= \frac{e^2h^2\sin^2\nu}{p^2} + \frac{h^2}{r^2} \\ &= \frac{h^2}{p^2}\left(1 + e^2 + 2e\cos\nu\right) \end{aligned}$$

Substituting in equation (6.14) and (6.9)

$$\begin{aligned} V^2 &= \frac{h^2}{p^2}\left(1 + \left[1 - \frac{p}{a}\right] + 2\left[\frac{p}{r} - 1\right]\right) \\ &= \frac{h^2}{p^2}\left(\frac{2}{r} - \frac{1}{a}\right) \\ &= \mathrm{GM}\left(\frac{2}{r} - \frac{1}{a}\right) && (6.16) \end{aligned}$$

6.2 Measured Parameters

The data collected from radar observation can be reduced to the following form.

A	azimuth of the apparent radiant.
z	zenith angle of the apparent radiant.
V_i	magnitude of the meteor's velocity in the atmosphere.
UT	Universal Time of the meteor observation.
MJD	Modified Julian Date of observation.

These parameters are based on a coordinate system centred on the observatory. The velocity vector of the meteor given the azimuth, zenith angle and speed as in Figure 4.38 is

$$\mathbf{v}_\mathrm{i} = -V_\mathrm{i}\begin{pmatrix} \sin z\cos A \\ \sin z\sin A \\ \cos z \end{pmatrix} \qquad (6.17)$$

6.3 Atmospheric Deceleration

Once the meteor starts to produce ionisation in the atmosphere it will begin to decelerate.[1] For an orbit reduction it is necessary to estimate the velocity, v_∞ of the meteor before it hits the atmosphere. The measured velocity, v will have been decelerated from v_∞. The classical method of deducing the atmospheric deceleration correction to the observed velocity can be found in McKinley (1961). The resistance or drag equation, (7-1) is

$$\frac{dv}{dt} = -\frac{\Gamma A}{m^{\frac{1}{3}}\rho_m^{\frac{2}{3}}}\rho_a v^2 \tag{6.18}$$

The differential mass equation, (7-2) gives

$$\frac{dm}{dt} = -\frac{\Lambda A}{2\zeta}\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}}\rho_a v^3 \tag{6.19}$$

The symbols used in these equations are defined below.

v	Velocity of the meteor	Γ	Drag coefficient, 0.5 $\longrightarrow$ 1.0
m	Mass of the meteor	A	Shape factor
ρ_m	Density of the meteor	Λ	Heat transfer coefficient, 0.6 $\longrightarrow$ 0.1
ρ_a	Air Density	σ	Ablation coefficient
ζ	Heat of ablation, 0.2 $\longrightarrow$ 1.6 $\times 10^{10}$ erg.gm^{-1}		

Differentiating equations (6.19) and (6.18) by parts, then integrating gives

$$\frac{dm}{dv} = \frac{\Lambda m^{\frac{2}{3}} m^{\frac{1}{3}} v}{2\Gamma\zeta} \tag{6.20}$$

$$\frac{dm}{m} = \frac{\Lambda}{2\Gamma\zeta} v.dv \tag{6.21}$$

$$\ln\left(\frac{m_\infty}{m}\right) = \frac{\Lambda}{4\Gamma\zeta}\left(v_\infty^2 - v^2\right) \tag{6.22}$$

where $v = v_\infty$ and $m = m_\infty$ outside the atmosphere. Defining the ablation coefficient and rearranging slightly gives

$$v_\infty^2 = v^2 + \frac{2}{\sigma}\ln\left(\frac{m_\infty}{m}\right) \tag{6.23}$$

$$\sigma = \frac{\Lambda}{2\Gamma\zeta} \tag{6.24}$$

The decrease in meteor mass as ablation proceeds is given by

$$m = m_\infty\left(1 - \frac{\rho_a}{3\rho_{a\ \max}}\right)^3 \tag{6.25}$$

[1]The atmosphere produces a frictional force on the interplanetary particle which is considerably greater than the accelerating gravitational influence of the Earth.

Where the atmospheric density at maximum ionisation is $\rho_a = \rho_{a\ \max}$. If the meteor velocity is measured at the position of maximum ionisation then classically

$$v_\infty^2 - v^2 = \frac{2}{\sigma} \ln \left(1 - \frac{\rho_{a\ \max}}{3\rho_{a\ \max}}\right)^{-3} \tag{6.26}$$

$$v_\infty^2 = v^2 + 1.216\frac{2}{\sigma} \tag{6.27}$$

It only remains to evaluate the ablation coefficient. The atmospheric density at maximum ionisation combines with equation (6.18) to give an expression for the deceleration at maximum ionisation (McKinley eqn 7-20).

$$\rho_{a\ \max} = \frac{3g\zeta \cos Z}{\Lambda A v^2} m_{\max}^{\frac{1}{3}} \rho^{\frac{2}{3}m} \tag{6.28}$$

$$\left(\frac{dv}{dt}\right)_{\max} = -\frac{3\Gamma\zeta \cos Z}{\Lambda \mathrm{H}} \tag{6.29}$$

$$= -\frac{3\cos Z}{2\sigma \mathrm{H}} \tag{6.30}$$

Making the assumption that for all meteors that the echo points occur at maximum ionisation will introduce some uncertainty. Moving the detection point up or down by half an atmospheric scale height (±3km) with respect to the height of maximum ionisation will affect the assumed deceleration so that for the classical case the meteor deceleration will be 15% too high or 30% too low. The error in the velocity correction will be less than 2km.s^{-1} in most cases.

A value for the ablation coefficient is discussed in Section 5.4. Substituting equation 5.8 into equation (6.27) provides a correction for the atmospheric deceleration of the meteor. The corrected velocity is labelled v_2.

$$v_2^2 = v_\infty^2 = v^2 + 0.80v^{1.6} \tag{6.31}$$

6.4 Rotation of the Earth

The velocity of the meteor should not depend on the rotation of the Earth. Effectively this involves shifting to a coordinate frame that is stationary with respect to the centre of the Earth. The corrected velocity of the meteor will be a vector sum of the observed velocity and the motion of the observing station.

$$\mathbf{v}_3 = \mathbf{v}_2 + \mathbf{v}_R \tag{6.32}$$

Figure 6.2 shows these vectors and defines the angles necessary to describe them. Figure 4.38 identifies the zenith and azimuth angles of a meteor. The observed velocity, $\mathbf{v}_2$ after correction for the atmospheric deceleration is

$$\mathbf{v}_2 = -V_2 \begin{pmatrix} \sin z \cos A \\ \sin z \sin A \\ \cos z \end{pmatrix} \tag{6.33}$$

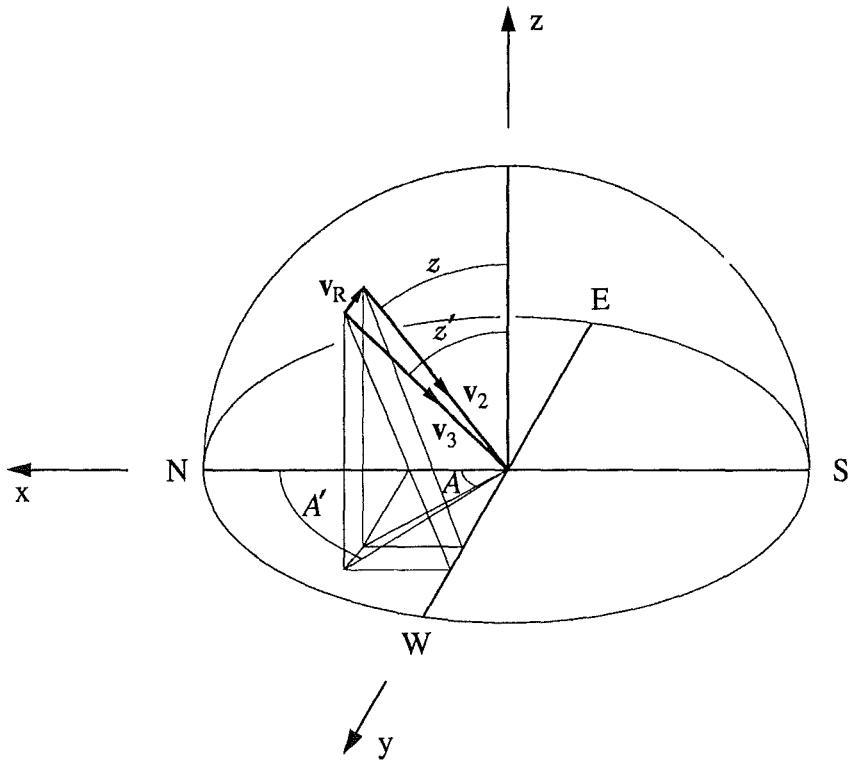


Figure 6.2: Vector diagram showing the effect of the Earth's rotation on the observed meteor velocity.

while the velocity of the meteor corrected for the Earth's rotation, $\bar{\mathbf{v}}_3$ is

$$\mathbf{v}_3 = -V_3 \begin{pmatrix} \sin z' \cos A' \\ \sin z' \sin A' \\ \cos z' \end{pmatrix} \tag{6.34}$$

The velocity of a point on the equator due to the Earth's rotation can be easily calculated. Given the Earth's equatorial radius, 6378.4 km and the number of seconds in a day the speed is approximately 0.4639 km.s^{-1}. Away from the equator this speed drops off as the cosine of the latitude. Allowing for the flattening of the Earth introduces a small modification of the resulting value. The flattening factor at 44° latitude is 0.99838 (Astronomical Almanac Explanatory Supplement, pp57-59). The velocity vector in km.s^{-1} for the motion of the observing station is

$$\mathbf{v}_R = -0.4631 \cos\phi \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{6.35}$$

The zenith angle and magnitude of the velocity are necessary to calculate the zenith attraction. V_3 is the magnitude of the velocity, $\mathbf{v}_3$, A' is the new azimuth angle and the zenith is z'

$$V_3 = \sqrt{\mathbf{v}_3 . \mathbf{v}_3} \tag{6.36}$$

$$\tan A' = \frac{(\mathbf{v}_3)_y}{(\mathbf{v}_3)_x} \tag{6.37}$$

$$\tan z' = \frac{\sqrt{(\mathbf{v}_3)_x^2 + (\mathbf{v}_3)_y^2}}{(\mathbf{v}_3)_z} \tag{6.38}$$

The correction of the radiant position to compensate for the Earth's rotation is called the diurnal aberration. Traditionally it has been done after the correction for zenith attraction and conversion to equatorial coordinates, see Porter (1952). The calculation of the zenith attraction is done in a geostationary frame based on the centre of the Earth. It is more natural to make this correction for the fact the observing station is rotating with respect to this geostationary frame before converting to it. The diurnal aberration affects both the zenith angle and scalar velocity used in Section 6.5. The small modification to the scalar velocity of the meteor due to rotation is neglected by both Porter and Lovell, although noted by McKinley (1961).

Both of these corrections follow naturally from the one vector sum, equation (6.32). The zenith angle and meteor speed are recalculated to facilitate the zenith attraction corrections.

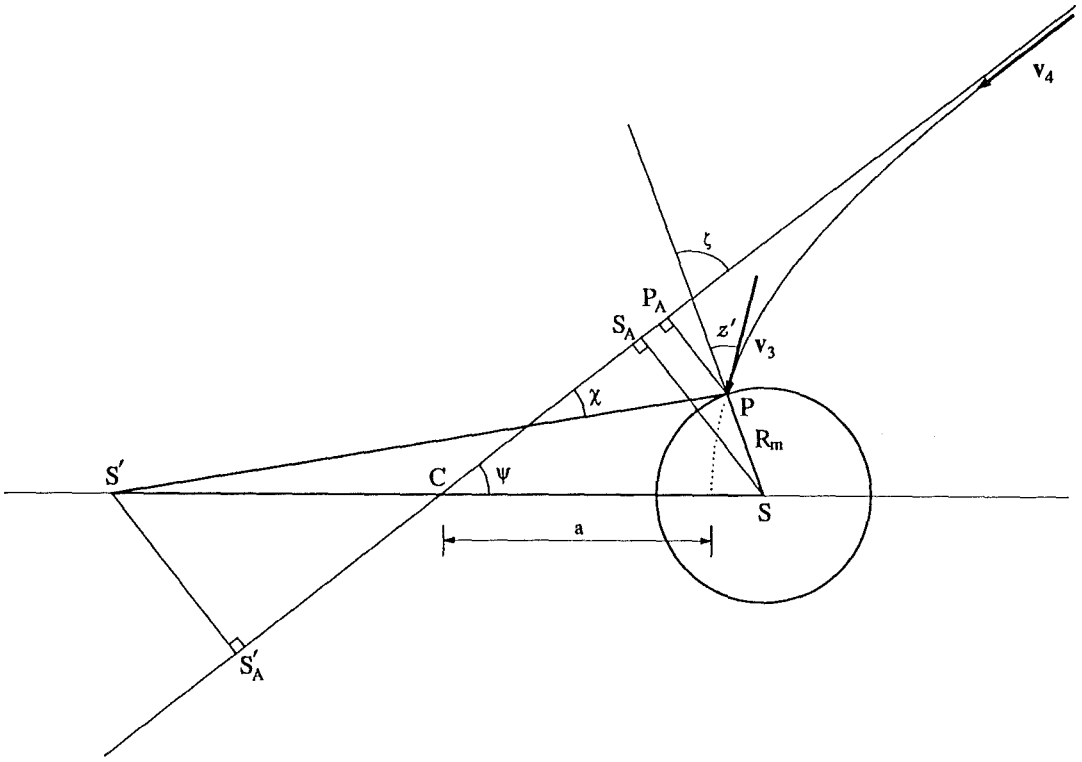


Figure 6.3: Motion of the meteoroid in the Earth's gravity field.

6.5 Acceleration in the Earth's gravitational field

At large distances from the Earth the meteoroid moves in an unperturbed heliocentric orbit with velocity $\mathbf{v}_4$. Before detection the meteoroid will traverse a hyperbolic path caused by the gravitational attraction of the Earth. The Earth's field will change both the speed and trajectory of the meteoroid. Figure 6.3 shows the perturbation due to the Earth's gravity well.

The speed change is calculated by considering energy conservation along the path. The change in direction is solely a shift in the zenith. This zenith correction is calculated using the geometric properties of the hyperbolic motion. It involves projecting into a plane that includes the meteor trajectory and the centre of the earth.

6.5.1 Increase in Speed

Outside the Earth's gravitational influence the potential energy of the meteoroid can be defined as zero. Effectively the meteoroid is at infinity. As it moves to the detected altitude it loses potential energy. $\frac{GM_\oplus m}{R_m}$ will be converted to kinetic energy where m is the mass of the meteor. So from energy

conservation the unperturbed velocity, V_4 can be found.

$$\tfrac{1}{2}mV_3^2 - \frac{\mathrm{GM}_\oplus m}{R_m} = \tfrac{1}{2}mV_4^2 \tag{6.39}$$

$$V_3^2 - \frac{2\mathrm{GM}_\oplus}{R_m} = V_4^2 \tag{6.40}$$

Meteors are observed to ablate about 100 km above the Earth's surface. The Earth has an equatorial radius of 6378.4 km with a flattening factor pertinent to 44° latitude of 0.99838. A reasonable radius for any observed meteor is therefore R_m = 6468 km. Numerical values of G = 6.67204 × 10^{-11} N.m^2.kg^{-2} and $\mathrm{M}_\oplus$ = 5.9742 × 10^{24} kg give for equation (6.40)

$$V_4 = \left(V_3^2 - 123.3\right)^{\frac{1}{2}} \tag{6.41}$$

where the velocity is in km.s^{-1}.

6.5.2 Zenith Attraction

In Figure 6.3 the unperturbed meteor trajectory is along the asymptote to its hyperbolic orbit. The undeviated zenith angle is therefore ζ. It can be calculated from the observed zenith angle, z' (corrected for the Earth's rotation) and the two velocities V_3 and V_4. This section does this by evaluating the zenith attraction correction.

$$\triangle z = \zeta - z' \tag{6.42}$$

The velocity of an orbiting particle given its position can be determined from equation (6.16). However for a hyperbolic trajectory the dimension a is negative. It is the distance from the periapse to the centre, C of the hyperbola. This distance was incorrectly labelled by Porter (1952) and the mistake repeated by Lovell (1954). They mistakenly took a = CS = CS′ although this did not affect their further derivations.

Both the final velocity of the meteor and the unperturbed velocity can be expressed as

$$V_3^2 = \mathrm{GM}_E\left(\frac{2}{R_m} + \frac{1}{a}\right) \tag{6.43}$$

$$V_4^2 = \frac{\mathrm{GM}_E}{a} \tag{6.44}$$

These equations can be rearranged giving

$$V_3^2 = V_4^2 a\left(\frac{2}{R_m} + \frac{1}{a}\right)$$

$$V_3^2 R_m = V_4^2\left(2a + R_m\right) \tag{6.45}$$

A geometrical argument based on properties of the hyperbolic orbit relates equation (6.45) to the zenith angles z' and ζ. The projection of the three sides of a triangle onto any straight line sums to zero. That is, projecting the triangle SPS′ in Figure 6.3 onto the asymptote gives

$$S'_A P_A - S_A P_A = S'_A S_A \tag{6.46}$$

The projection of PS onto the asymptote is

$$P_A S_A = R_m \cos\zeta \tag{6.47}$$

A hyperbola can be defined as the locus of points such that $S'P - SP = 2a$. The length S′P is therefore $(2a + R_m)$. The projection of S′P will depend on the angle χ. For a hyperbola the tangent at any point bisects the angle subtended by the two foci to that point. The angle $\widehat{S'PS}$ is thus $2z'$. Considering the triangle containing χ, (Figure 6.3)

$$\chi + \zeta + (\pi - 2z') = \pi \tag{6.48}$$

and so $\chi = 2z' - \zeta$. The length $S'_A P_A$ is therefore

$$P_A S'_A = (2a + R_m)\cos(2z' - \zeta) \tag{6.49}$$

The distance between the two foci of a hyperbola is $S'S = 2ae$ where its eccentricity is $e = 1/\cos\Psi$. The projection onto the asymptote of S′S is

$$S'_A S_A = S'S\cos\Psi = 2a \tag{6.50}$$

The total sum from equation (6.46) is hence

$$(2a + R_m)\cos(2z' - \zeta) - R_m\cos\zeta = 2a \tag{6.51}$$

Adding R_m to both sides and rearranging gives

$$R_m(1 - \cos\zeta) = (2a + R_m)(1 - \cos(2z' - \zeta)) \tag{6.52}$$

Comparing with equation (6.45) gives

$$V_4^2(1 - \cos\zeta) = V_3^2(1 - \cos(2z' - \zeta)) \tag{6.53}$$

Now $\Delta z = \zeta - z'$ so,

$$\begin{aligned} \zeta &= z' + \Delta z \\ 2z' - \zeta &= z' - \Delta z \end{aligned} \tag{6.54}$$

Substituting these angles in equation (6.53) and using the relation $\cos A = 1 - 2\sin^2(A/2)$ gives

$$\begin{aligned} V_4^2\left(1 - \left\{1 - 2\sin^2\frac{z' + \Delta z}{2}\right\}\right) &= V_3^2\left(1 - \left\{1 - 2\sin^2\frac{z' - \Delta z}{2}\right\}\right) \\ V_4^2\sin^2\tfrac{1}{2}(z' + \Delta z) &= V_3^2\sin^2\tfrac{1}{2}(z' - \Delta z) \end{aligned} \tag{6.55}$$

Expanding the sine expressions, dividing by $\cos\frac{1}{2}\triangle z \cos\frac{1}{2}z'$ then rearranging gives the zenith correction in terms of the magnitude of the measured velocity, its zenith angle and the unperturbed orbital speed.

$$\begin{aligned} V_4\left\{\sin\frac{z'}{2}\cos\frac{\triangle z}{2}+\cos\frac{z'}{2}\sin\frac{\triangle z}{2}\right\} &= V_3\left\{\sin\frac{z'}{2}\cos\frac{\triangle z}{2}-\cos\frac{z'}{2}\sin\frac{\triangle z}{2}\right\} \\ V_4\left\{\tan\frac{z'}{2}+\tan\frac{\triangle z}{2}\right\} &= V_3\left\{\tan\frac{z'}{2}-\tan\frac{\triangle z}{2}\right\} \\ \tan\left(\tfrac{1}{2}\triangle z\right) &= \frac{V_3-V_4}{V_3+V_4}\tan\left(\tfrac{1}{2}z'\right) \end{aligned} \tag{6.56}$$

The unperturbed zenith angle therefore follows from the measured zenith angle and the zenith correction, equation (6.42). The observed zenith angle is always in the range $0 < z < \frac{\pi}{2}$ and that $V_3 > V_4$. This means that $\tan(\frac{1}{2}\triangle z)$ is always positive. That is, the acceleration due to the Earth's field always moves the radiant toward the zenith, hence the term 'zenith attraction'.

The velocity vector of the meteoroid outside the Earth's field in terms of our topocentric coordinates is therefore

$$\mathbf{v}_4 = -V_4\begin{pmatrix}\sin(z'+\triangle z)\cos A\\ \sin(z'+\triangle z)\sin A\\ \cos(z'+\triangle z)\end{pmatrix} \tag{6.57}$$

6.6 Coordinate Transformations

So far we have assumed the centre of the Earth is fixed in space. It is necessary to remove the velocity component due to the Earth's orbital motion around the Sun. This is most conveniently done in heliocentric ecliptic coordinates. It is evaluated in the next section. This section breaks the transformation to these coordinates into a series of straight forward rotations. Each is described by a 3×3 rotation matrix.

Figure 6.4 shows a rotation of angle θ about the z axis. A vector $\mathbf{r}$ in the unprimed frame may be converted to $\mathbf{r}'$ in the primed frame by

$$\mathbf{r}' = \mathrm{A}\mathbf{r} \tag{6.58}$$

where A is the rotation matrix

$$\mathrm{A} = \begin{pmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{pmatrix} \tag{6.59}$$

The rows of the matrix A are just the position vectors of the new primed unit vectors in terms of the old unprimed frame. Establishing the rotation matrix is a simple matter of describing the new unit vectors in terms of the old system. A series of rotations can then be conducted by successively multiplying by the respective matrices.

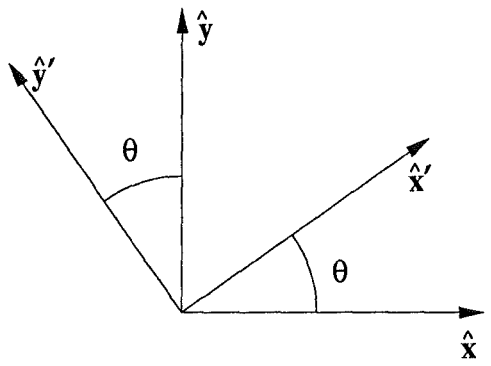


Figure 6.4: A rotation of θ about the z axis in the x-y plane.

6.6.1 Conversion to Equatorial Coordinates

Equatorial coordinates are based on the centre of the Earth. The z axis is directed toward the north celestial pole and the x axis toward the vernal equinox. The y axis forms the third component of a right handed set. Positions in terms of this set of coordinates are usually referred to by radius, declination and right ascension. It is more straight forward however to leave the velocity as x,y,z components.

First rotate the local north axis toward the north celestial pole. This is just a rotation of the latitude angle as shown in Figure 6.5. It is effected by the matrix A_1

$$A_1 = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix} \tag{6.60}$$

The latitude of Birdlings Flat is, ϕ = -43.825°(north positive). The velocity vector referred to this new set of axes is

$$\mathbf{v}' = A_1\mathbf{v} \tag{6.61}$$

Transferring the axes around while still maintaining a right handed set is also shown in Figure 6.5. Using a matrix T to do this gives $\mathbf{v}'' = T\mathbf{v}'$, where

$$T = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \tag{6.62}$$

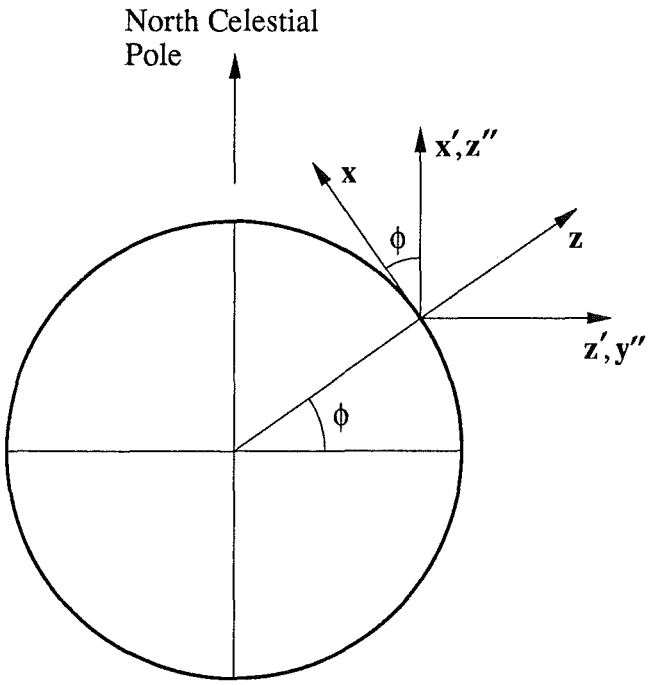


Figure 6.5: Rotate the local north axis toward the north celestial pole. Then transfer the axes so z'' points to the North Celestial Pole. x'' is out of the page.

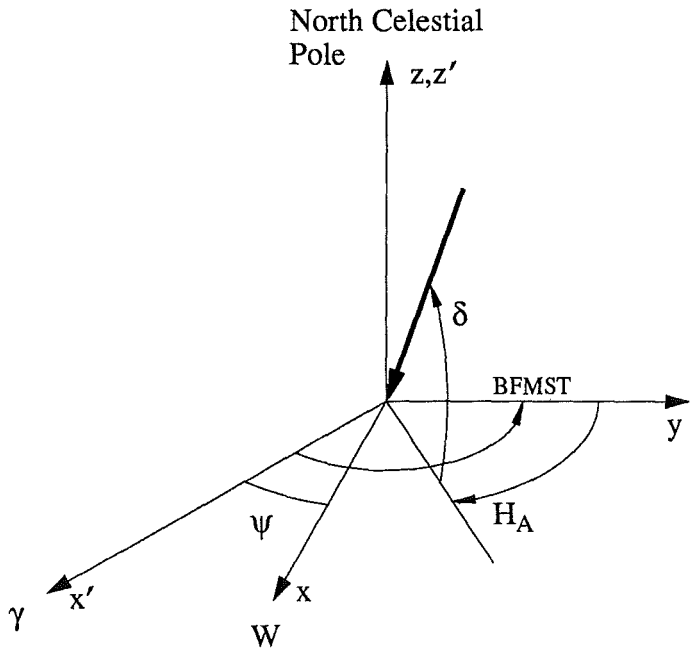


Figure 6.6: Rotate the x axis toward the vernal equinox. This produces an equatorial coordinate set.

The declination, δ and hour angle, H_A are shown in the diagram. With this series of matrix rotations it is not necessary to calculate them explicitly.

The new x axis needs to be directed at the vernal equinox, Υ. Figure 6.6 shows the rotation necessary. The longitude of the radar fixes the meridian; that is the yz plane. The angle between this plane and the vernal equinox is by definition the mean sidereal time of the radar station, BFMST (Birdlings Flat Mean Sidereal Time).

Given the modified Julian date (MJD) and the Universal Time (UT) of the observation, the Greenwich Mean sidereal time, GMST can be calculated (Astronomical Almanac, 1989). The variable d' defines the days since 1989 January 0.

$$\text{UT} = \text{NZST} - 12 \tag{6.63}$$

$$\text{MJD} = 47526.0 + d' + \frac{\text{UT}}{24} \tag{6.64}$$

$$\text{GMST} = 6.424454 + 0.0657098243\ d' + 1.00273791\text{UT} \tag{6.65}$$

Adding the east longitude of the radar station, then adding or subtracting multiples of 24 hours gives the BFMST in hours.

$$\text{BFMST} = (\text{GMST} + 11.512444)_{\text{mod } 24} \tag{6.66}$$

The rotation angle ψ is now known and the rotation matrix follows from this,

$$\psi = \text{BFMST} - 90^\circ \tag{6.67}$$

$$\mathrm{A}_2 = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.68}$$

The conversion of a vector $\mathbf{r}$ in topocentric coordinates to a description of the vector $\mathbf{r}'''$ in equatorial coordinates is

$$\underset{\text{equatorial}}{\mathbf{r}'''} = \underset{\text{topocentric}}{\mathrm{A}_2\mathrm{T}\mathrm{A}_1\mathbf{r}} \tag{6.69}$$

The right ascension , α and declination, δ of the radiant at this stage of the correction process could be calculated knowing that the velocity in equatorial coordinates is

$$\mathbf{v}_5 = \mathrm{A}_2\mathrm{T}\mathrm{A}_1\mathbf{v}_4 \tag{6.70}$$

$$= -V_5 \begin{pmatrix} \cos\delta\cos\alpha \\ \cos\delta\sin\alpha \\ \sin\delta \end{pmatrix} \tag{6.71}$$

Here this is an unnecessary step in calculating the orbital parameters of the meteoroid. The method described by Porter (1952) needs to calculate them as an intermediate step.

6.6.2 Conversion to Heliocentric Coordinates

Ecliptic coordinates are fixed by placing the Earth's orbit in the xy plane. The x axis is directed toward the vernal equinox, Υ. The heliocentric frame used here places the y axis on the line from the Earth to the Sun. A simple translation can then put the origin of the coordinate system at the Sun's centre. In this section the unprimed axes are equatorial coordinates.

First rotate the z axis to be normal to the Earth's orbit. This is a rotation of the Earth's inclination, χ about the x axis, Figure 6.7. The matrix A_3 will effect this rotation

$$\mathrm{A}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\chi & \sin\chi \\ 0 & -\sin\chi & \cos\chi \end{pmatrix} \tag{6.72}$$

$$\underset{\text{ecliptic}}{\mathbf{r}'} = \underset{\text{equatorial}}{\mathrm{A}_3\mathbf{r}} \tag{6.73}$$

The angle between the x axis and the line from the Earth to the Sun is called the longitude of the Sun, LS (see Figure 6.8). It can be evaluated

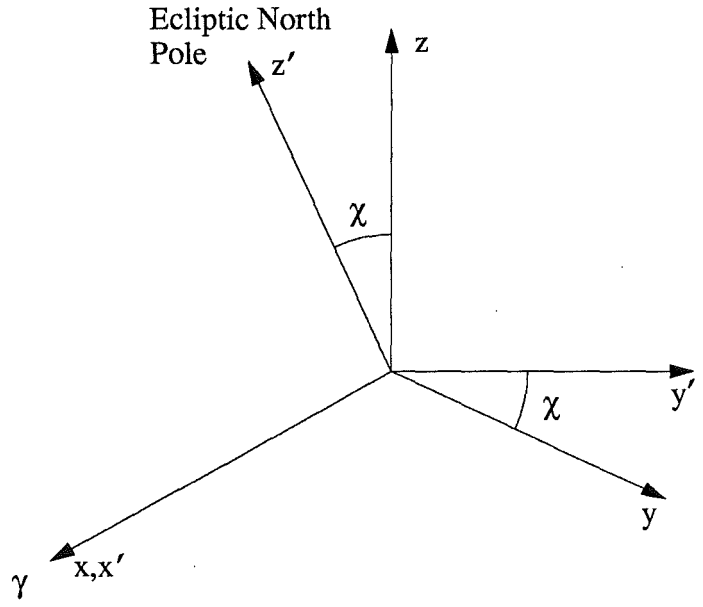


Figure 6.7: Rotate the Earth's axis to be normal to the ecliptic so converting to ecliptic coordinates.

empirically using the *Low precision formulas for the Sun's coordinates*, Astronomical Almanac (1989) page C24. The mean longitude of the sun, MS and the longitude on the mean epoch of date, LS can be calculated.

$$n = -4018.5 + (\text{days since 1989 Jan 0}) + \frac{\text{UT}}{24} \tag{6.74}$$

$$\text{MS} = 357.^\circ528 + 0.9856003n \tag{6.75}$$

$$\text{LS} = 280.^\circ460 + 0.^\circ985647n + 1.915 \sin \text{MS} + 0.020 \sin 2\text{MS} \tag{6.76}$$

for the year 1987 and $0 < \text{LS} < 360^\circ$.

We need to rotate the y axis to point at the Sun. A rotation of ψ about the z axis is

$$\psi = \text{LS} - 90^\circ \tag{6.77}$$

$$\text{A}_4 = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.78}$$

$$\mathbf{r}'' = \text{A}_4\mathbf{r}' \tag{6.79}$$

heliocentric ecliptic

A complete conversion of the velocity vector, $\mathbf{v}_4$ in topocentric coordinates

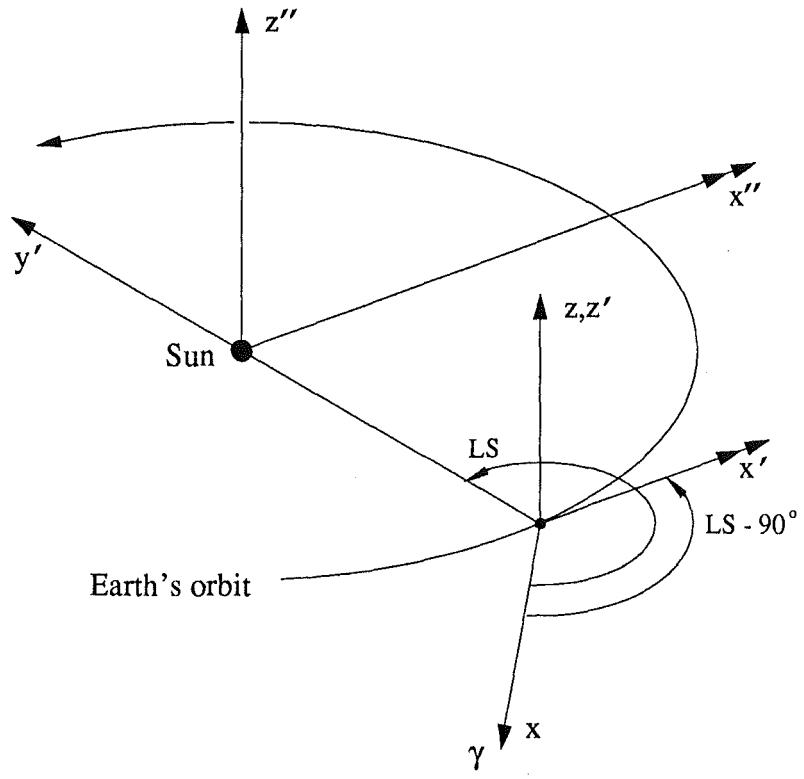


Figure 6.8: Final rotation to heliocentric ecliptic coordinates.

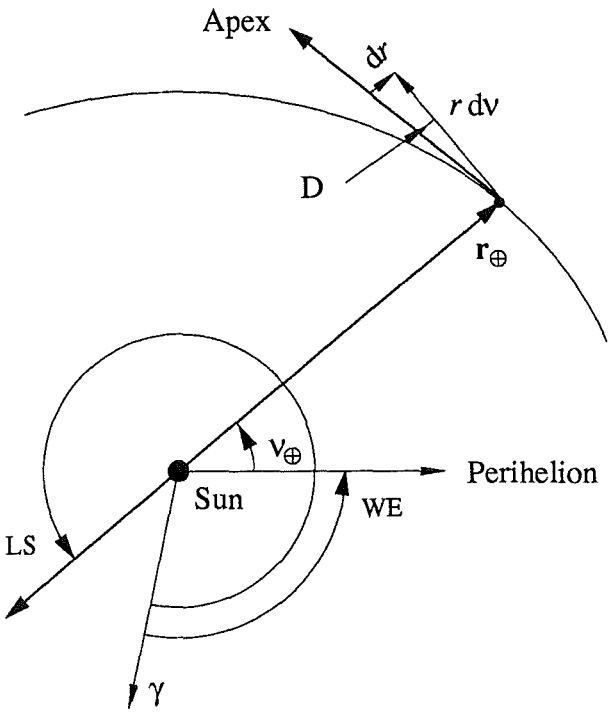


Figure 6.9: The orbital motion of the Earth in heliocentric coordinates.

to $\mathbf{v}_6$ in heliocentric coordinates is

$$\mathbf{v}_6 = A_4 A_3 A_2 T A_1 \mathbf{v}_4 \tag{6.80}$$

6.7 Orbital Motion of the Earth

The velocity vector, $\mathbf{v}_6$ of the meteor still depends on the Earth's orbital motion through space. The velocity of the observing platform (the Earth) needs to be subtracted from $\mathbf{v}_6$ to get the true space velocity of the meteor. Since the Earth's orbit is slightly elliptical its orbital speed will vary at different points in the orbit. This small correction is neglected by both Porter and Lovell. Figure 6.9 diagrammatically describes the Earth's velocity in heliocentric coordinates.

The angle D, the deviation of the Apex from perpendicular to the solar Longitude is by definition

$$\tan \mathrm{D} = -\frac{1}{r}\frac{\partial r}{\partial \nu} \tag{6.81}$$

By substituting the expression for the radius, equation (6.9) and semi-major

axis, equation (6.14) of a particle in a central field this becomes

$$\tan \mathrm{D} = \frac{-e \sin \nu}{1 + e \cos \nu} \tag{6.82}$$

The Earth's true anomaly, $\nu_\oplus$, can be calculated empirically Astronomical Almanac (1989) page C24. The longitude of the Sun at the time of observation is given by equation (6.76). The Earth's longitude of perihelion (WE) and hence the true anomaly (TE) or $\nu_\oplus$ follow from this. The day variable d is defined by equation (6.83).

$$\begin{aligned} d &= \text{days since 1989 Jan 0} \\ &\quad + \text{ fraction of day from zero hours UT} && (6.83) \\ \mathrm{WE} &= 102.^\circ 749176 + 0.000047068d && (6.84) \\ \mathrm{TE} &= \mathrm{LS} + 180^\circ - \mathrm{WE} && (6.85) \end{aligned}$$

The orbital radius of the Earth is just the distance of the sun from the Earth (RS).

$$\begin{aligned} \mathrm{RS} &= 1.00014 - 0.01671 \cos \mathrm{MS} - 0.00014 \cos 2\mathrm{MS} && (6.86) \\ r_\oplus &= \mathrm{RS} && (6.87) \end{aligned}$$

The orbital speed of the Earth can now be calculated using its orbital radius.

$$V_\oplus^2 = \frac{\mathrm{GM}}{1 \text{ A.U.}} \left(\frac{2.00}{r_\oplus} - 1.00 \right) \tag{6.88}$$

The Earth's velocity vector, $\mathbf{v}_\oplus$ is defined in terms of D and the orbital speed, $V_\oplus$. The meteor's true velocity can be calculated using this.

$$\begin{aligned} \mathbf{v}_\oplus &= V_\oplus \begin{pmatrix} \cos \mathrm{D} \\ \sin \mathrm{D} \\ 0 \end{pmatrix} && (6.89) \\ \mathbf{v}_m &= \mathbf{v}_6 + \mathbf{v}_\oplus && (6.90) \end{aligned}$$

The various effects of observing the meteor velocity from the Earth have been eliminated. To calculate the orbital elements for the meteoroid we still need to know the position vector corresponding to this unperturbed velocity.

6.8 Heliocentric Position of the Meteor

The heliocentric position of an observed meteor can most simply be described as a vector sum. This sum includes: The earth's orbital radius at the time of observation (from the centre of the Sun to the centre of the Earth), the radius of the Earth (from the centre of the Earth to the observing station) and the radar range (from the observing station to the meteor trail). This position

vector is that of the observed meteor and corresponds with the observed velocity vector $\mathbf{v}_1$. We, however, require the position vector of the unperturbed heliocentric velocity of the meteor, $\mathbf{v}_m$. That is, the location the meteoroid would have had, with a velocity $\mathbf{v}_m$ if it were unperturbed by the Earth's presence.

Of the four corrections made to the velocity (atmospheric deceleration, Earth's rotation, gravitational attraction, Earth's orbital motion) only the Earth's gravitational attraction will significantly change the meteoroid position vector. Arguing from Figure 6.3 the contributions of the Earth's radius and the radar range should be replaced by the distance SS_A, the impact parameter of the velocity.

Porter (1952) p84-85 concludes that to hit the Earth in general meteoric particles must approach within a few thousand kilometers. Exceptionally low velocity orbits could extend this to several tens of Earth radii. However sufficiently slow meteoric particles of this type do not produce enough ionisation to be readily detected as radio meteors.

Detailed consideration of the meteor's impact parameter is however unnecessary. It is negligible in comparison with the Earth's orbital radius. The Earth's distance from the Sun is calculated within the reduction code using the empirical expression, equation (6.86) from the Astronomical Almanac. An alternative expression for the distance, $r_\oplus$, using the parameters of the Earth's orbit is given below. The orbital eccentricity of the Earth has usually been neglected in calculating meteor orbits. The position vector that corresponds to the meteor velocity $\mathbf{v}_m$ is

$$\mathbf{r}_m = \mathbf{r}_\oplus = \begin{pmatrix} r_\oplus \\ 0 \\ 0 \end{pmatrix} \tag{6.91}$$

$$r_\oplus = \frac{1 - e^2}{1 + e\cos\nu} \tag{6.92}$$

6.9 Orbital Elements

The orbit of a meteoroid can be defined in terms of five orbital elements: Longitude of the ascending node (Ω), inclination (i), argument of perihelion (ω), eccentricity (e) and its semi-major axis (a). A sixth element, the time of perihelion passage locates the position of the particle in this orbit. The position and velocity of a meteoroid uniquely defines the set of orbital elements that describe its orbit. We need to calculate the orbital elements in terms of this information.

The longitude of the ascending node and the inclination locate the plane of the orbit in space. Figure 6.10 provides definitions for these two quantities. Where the z component of $\mathbf{v}_m$ (v_z) is positive the Earth is at the ascending node. Otherwise the meteor is at the descending node and Ω is 180° further

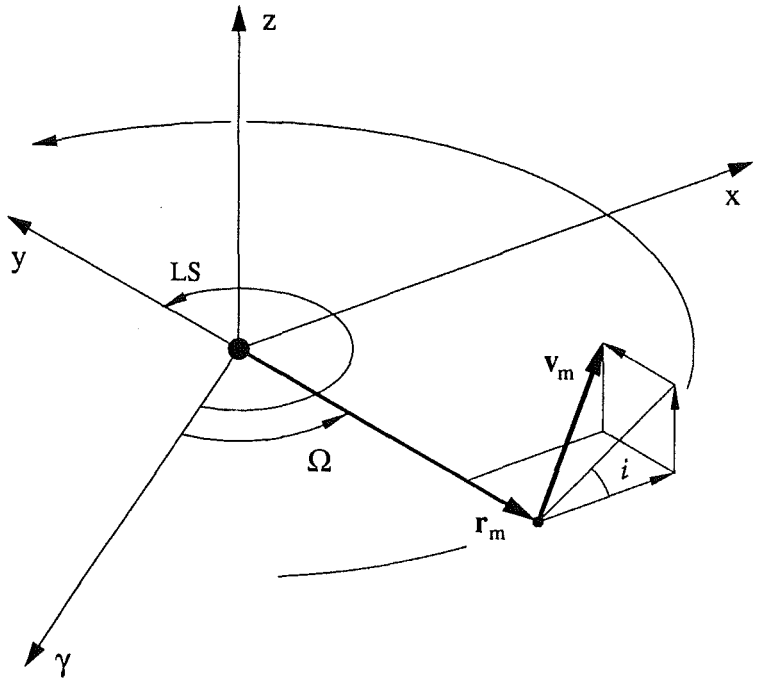


Figure 6.10: The observation data can be reduced to the heliocentric position and corrected velocity of the meteoroid.

around the orbit. The longitude of the ascending node is in the range 0 to 360°.

$$\Omega = \begin{cases} \text{LS} - 180° & \text{v}_z > 0 \\ \text{LS} & \text{v}_z < 0 \end{cases} \tag{6.93}$$

This longitude is that corresponding to the mean epoch of date. Most solar system minor body orbits are given in terms of the standard epoch of 1950.0. The precession since 1950.0 is calculated and subtracted to give the longitude of the ascending node for the standard epoch (Ω_{1950}).

$$\text{Julian century } (J) = \frac{\text{MJD} - 33281.923}{36534.22} \tag{6.94}$$

$$\text{Precession} = 1.396319\ J\ + 0.0003083\ J^2 \tag{6.95}$$

$$\Omega_{1950} = \Omega - \text{Precession} \tag{6.96}$$

The inclination is the angle between the orbital planes of the earth and the meteoroid. The inclination ranges from 0 to 180°and refers to the angle measured at the ascending node. Values greater than 90° indicate retrograde motion. It follows simply from

$$\tan i = \frac{\text{v}_z}{\text{v}_x} \tag{6.97}$$

Ascending node,	$\text{v}_z > 0$.	
$\text{v}_x < 0$	retrograde	$90° < i < 180°$
$\text{v}_x > 0$	prograde	$0° < i < 90°$
Descending node,	$\text{v}_z < 0$.	
$\text{v}_x < 0$	prograde	$0° < i < 90°$
$\text{v}_x > 0$	retrograde	$90° < i < 180°$

The final three elements deal with the position and shape in the plane of the orbit. Figure 6.11 identifies these elements in addition to the semi-latus rectum, p and the true anomaly, ν. To evaluate these we first calculate p. The two components of the velocity in the plane of the orbit follow from Figure 6.10.

$$\text{v}_r = -\text{v}_y \tag{6.98}$$

$$\text{v}_\nu^2 = \text{v}_x^2 + \text{v}_z^2 \tag{6.99}$$

Using the aerial velocity, equation (6.8) and the motion in a central gravitational field, equation (6.11) the semi-latus rectum can be found.

$$h^2 = r_m^2 \text{v}_\nu^2 \tag{6.100}$$

$$p = \frac{h^2}{\text{GM}_\odot} \tag{6.101}$$

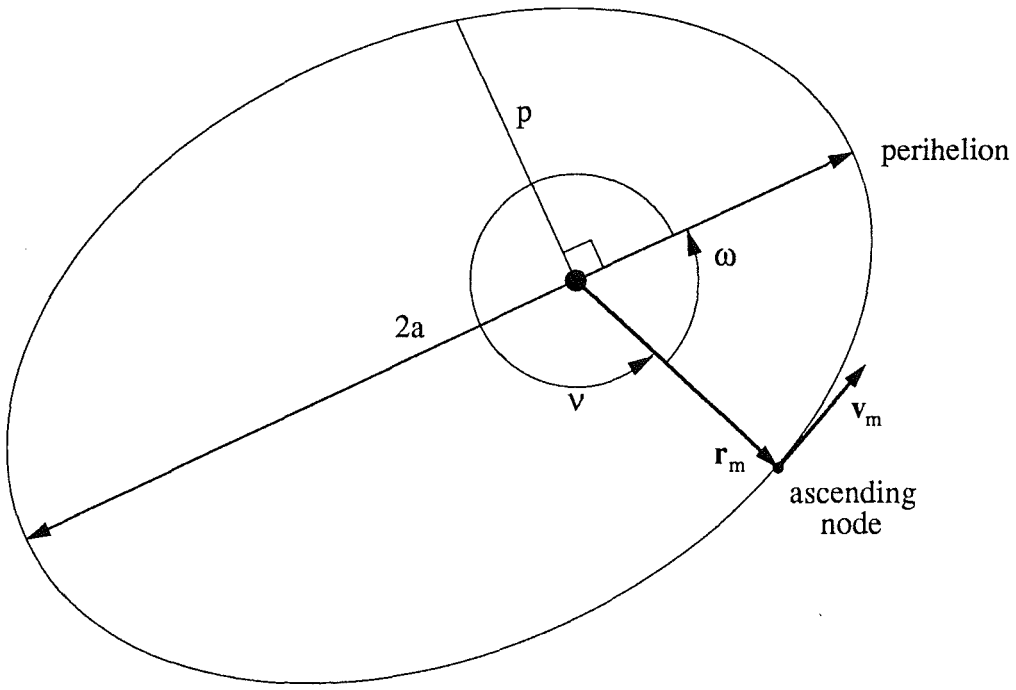


Figure 6.11: Orbital motion of a meteoroid detected at the ascending node. For detection at the descending node ω is measured in the same sense but from the ascending node, 180° further around the orbit.

The solar mass is 3.32958×10^5 that of the Earth. The Earth's mass is 5.9742×10^{24} kg. Rearranging and combining equations (6.9) and (6.10) for a general conic gives a ratio for the true anomaly

$$\tan \nu = \frac{v_r \sqrt{p/GM_\odot}}{p/r - 1} \tag{6.102}$$

Since ν is measured from the perihelion for v_r positive, ν will range from 0 to 180°. Where v_r is negative, $180° < \nu < 360°$. The argument of perihelion follows from this. Its actual value depends on whether the meteor was detected at the ascending or descending node.

$$\omega = \begin{cases} 360° - \nu & v_z > 0 \\ 180° - \nu & v_z < 0 \end{cases} \tag{6.103}$$

Using the equations for a conic section the eccentricity follows from equation (6.9) and the semi-major axis from equation (6.14). The perihelion distance, q is calculated from equation (6.9) with $\nu = 0$.

$$e = \left(\frac{p}{r} - 1\right) \frac{1}{\cos \nu} \tag{6.104}$$

$$a = \frac{p}{1 - e^2} \tag{6.105}$$

$$q = \frac{p}{1 + e} \tag{6.106}$$

6.10 Orbit Calculation Example

The orbit calculation in this appendix provides an illustration of the new orbit finding technique. The radar station must be able to determine the position of the apparent radiant and the speed of the meteor in the atmosphere. For the Canterbury radar, AMOR, this information is calculated from the time of observation, two timelags between receiver sites and the elevation angle of the echo at the Home site. The more general radiant and speed data has been calculated and is also presented as a possible starting point for the calculation.

The actual meteor parameters used in the example match those of an η Aquarid shower meteor. Station data have been invented to reflect a meteor observed at the local transit of the shower radiant.

Station Data

Observation	1990 May 4	0806:00
Lag12 (t_1 - t_2)	16 pulses	0.0422 sec
Lag13 (t_1 - t_3)	0 pulses	0.0000 sec
Elevation	42.0 deg	

Measured Parameters (6.2)

1990 May 4	0806:00 NZST
3.0°	azimuth angle
42.0°	zenith angle
62.12	meteor speed (km.s^{-1})

The observed velocity vector in the station frame of reference is therefore

$$\mathbf{v}_1 = \begin{pmatrix} -41.54 \\ -2.18 \\ -46.14 \end{pmatrix}$$

Atmospheric Deceleration (6.3)

The velocity above the atmosphere. $V_2 = 66.72$km.s^{-1}.

$$\mathbf{v}_2 = \begin{pmatrix} -44.61 \\ -2.34 \\ -49.55 \end{pmatrix}$$

Rotation of the Earth (6.4)

Velocity with respect to a non-rotating Earth. $V_3 = 66.71$km.s^{-1}.

$$\mathbf{v}_R = \begin{pmatrix} 0.00 \\ -0.33 \\ 0.00 \end{pmatrix}, \qquad \mathbf{v}_3 = \begin{pmatrix} -44.61 \\ -2.00 \\ -49.55 \end{pmatrix}$$

Earth's gravitational acceleration (6.5)

The corrected azimuth, zenith and unperturbed speed are calculated in this step. The velocity vector after the increase in speed and the change in the zenith angle has been allowed for. $V_4 = 65.77$km.s^{-1}.

65.77	Unperturbed Speed (km.s^{-1})
2.57°	Azimuth (corrected)
42.34°	Zenith (corrected)

$$\mathbf{v}_4 = \begin{pmatrix} -44.25 \\ -1.99 \\ -48.62 \end{pmatrix}$$

Coordinate Transformations (6.6)

Rotate coordinate set to the north celestial pole

$$\begin{pmatrix} 0.7215 & 0.0000 & -0.6925 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.6925 & 0.0000 & 0.7215 \end{pmatrix} \begin{pmatrix} -44.25 \\ -1.99 \\ -48.62 \end{pmatrix} = \begin{pmatrix} +1.74 \\ -1.99 \\ -65.72 \end{pmatrix}$$

Point z axis to the north celestial pole

$$\begin{pmatrix} 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \\ 1.0000 & 0.0000 & 0.0000 \end{pmatrix} \begin{pmatrix} +1.74 \\ -1.99 \\ -65.72 \end{pmatrix} = \begin{pmatrix} -1.99 \\ -65.72 \\ +1.74 \end{pmatrix}$$

Greenwich day number	Day	=	123
Universal time	UT	=	20.10 hrs
modified Julian date	MJD	=	48014.8375
Greenwich mean sidereal time	GMST	=	10.864 hrs
Longitude of Birdlings Flat radar	Long.	=	11.512 hrs
Birdlings Flat mean sidereal time	BFMST	=	22.376 hrs

Rotate x axis to vernal equinox, v_5

$$\begin{pmatrix} -0.4124 & 0.9110 & 0.0000 \\ -0.9110 & -0.4124 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{pmatrix} \begin{pmatrix} -1.99 \\ -65.72 \\ +1.74 \end{pmatrix} = \begin{pmatrix} -59.05 \\ +28.91 \\ +1.74 \end{pmatrix}$$

65.77	Unperturbed Speed (km.s^{-1})
333.9°	Right ascension (corrected)
-1.5°	Declination (corrected)

Rotate to ecliptic coordinates

$$\begin{pmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.9174 & 0.3979 \\ 0.0000 & -0.3979 & 0.9174 \end{pmatrix} \begin{pmatrix} -59.05 \\ +28.91 \\ +1.74 \end{pmatrix} = \begin{pmatrix} -59.05 \\ +27.22 \\ -9.91 \end{pmatrix}$$

Longitude of the sun (mean epoch of date) $L_\odot = 43.12°$.
Mean anomaly of the sun $MS = 118.69°$.

Rotate to heliocentric coordinates v_6

$$\begin{pmatrix} 0.6835 & -0.7299 & 0.0000 \\ 0.7299 & 0.6835 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{pmatrix} \begin{pmatrix} -59.05 \\ +27.22 \\ -9.91 \end{pmatrix} = \begin{pmatrix} -60.23 \\ +24.50 \\ -9.91 \end{pmatrix}$$

Orbital Motion of the Earth (6.7)

The Earth's velocity at the time of the observation is $V_\oplus = 29.54\text{km.s}^{-1}$. The Earth's velocity is subtracted from the meteor velocity to give the final heliocentric meteoroid velocity ($V_m = 40.76\text{km.s}^{-1}$).

Earth's true anomaly	$\nu_\oplus$	=	120.3°
Deviation of the Apex	D	=	-0.0145°

$$\mathbf{v}_\oplus = \begin{pmatrix} 29.54 \\ -0.43 \\ 0.00 \end{pmatrix} \qquad \mathbf{v}_m = \begin{pmatrix} -30.69 \\ -24.93 \\ -9.91 \end{pmatrix}$$

Heliocentric Position of the Meteor (6.8)

The position vector corresponding to the meteoroid's velocity vector is assumed to be just that of the Earth at that time. $R_m = R_\oplus = 1.008$ A.U.

Orbital Elements (6.9)

Precession from 1950		=	0.563°
Longitude of ascending node (1950)	Ω_{1950}	=	42.56°
Inclination of orbital plane	i	=	162.1°
Semi latus rectum	p	=	1.192 A.U.
True anomaly	ν	=	78.7°
Argument of perihelion	w	=	101.3°
Eccentricity of the orbit	e	=	0.932
Semi-Major axis	a	=	9.05 A.U.
Perihelion distance	q	=	0.617 A.U.

Chapter 7

Eta Aquarid Meteor Stream

7.1 Introduction

The association of the Orionid and η Aquarid meteor showers with the orbit of Comet Halley is well recognised. The 1986 apparition of the Comet provided an incentive to observe the streams for any signs of enhanced meteor activity. The radiant point for the northern hemisphere Orionid shower is too far north to be observed by radar from Christchurch. The slightly southern declination of the η Aquarid shower permits its detection.

Other features of the stream make it ideal for testing the radar. Its relatively large geocentric velocity ($65 km.s^{-1}$) produces higher ionisation densities improving the chance of detection by the radar. The retrograde orbit provides a distinctive enhancement at the 165° inclination of the stream. The only planetary close encounter is with the Earth so the loss rate from the stream is relatively low. The long period of the orbit means the particles hit the Earth at near the heliocentric escape velocity. This provides a fairly precise check on the accuracy of the velocity determinations. The high velocity means this test is done for orbits where the radar is most sensitive to timing errors.

The need to have an observing system operational by May each year was a major prompt for equipment development. By 1990 the η Aquarid observing run had become a major theme of the PhD thesis and that stream was chosen to provide an astronomical calibration of the orbit radar. In fact the evolution of the radar system can be directly traced in the equipment used in the annual observations of this shower (see Table 7.1).

Very few orbits for meteors in the η Aquarid shower have been measured by stations in the northern hemisphere. The radiant is not very well placed for visual observations. Lindblad (1989) notes that most authors use the mean orbit from Cook's list (Cook, 1973) which is based on only one meteor from the Harvard survey. Lindblad reviews the orbital elements for those meteors identified as belonging to the η Aquarid shower. These come from a variety of sources, the Harvard Super Schmidt program (7), Nippon Meteor Society

1987	Eye-ball observations of meteor echoes off an oscilloscope screen. Battle Ship style recording ('C7 . . G5 C4 . . . ') by hand to give meteor rates and range information.
1988	Microprocessor daisy chain system and Apple IIe data logger. Raw data recorded on 10 floppy disks during the night because the magnetic tape interface had been stolen in a third burglary. The system was rebuilt in a *mobile* configuration.
1989	'Luggable' PC (IBM AT-compatible) detection of meteors and control of the radar. This development version waited until one meteor had been detected then stopped and stored the observation record.
1990	Final observation program handling simultaneous meteors. This system identifies and provides timing for persistent trails.
1991	Observing experience and hardware changes allowed the system to run itself, unattended, for days at a time. Continuous observations from April 27 to May 16 were successful.

Table 7.1: AMOR η Aquarid Observing History.

observations (4) and the Harvard Radio meteor project (5) for a total of 16 orbits up to 1988. Additional observations are needed to improve the statistical significance of the mean orbital elements for the η Aquarid stream.

With the help of ancient Chinese observations of Halley's comet Yeomans and Kiang (1981) have integrated the comet's orbit back to 1404 BC. The orbital elements for the 989 AD apparition of the Comet Halley agree well with Lindblad's mean orbit for the η Aquarid shower. McIntosh and Hajduk (1983) discuss a new model for the η Aquarid meteor stream. They note that the nodes of the comet orbit cross the Earth's orbit between 530-607 AD at the position of the η Aquarids. The most recent meteoric material that comprises the η Aquarid stream was probably ejected from Comet Halley at some time during perihelion passage in the eight apparitions from 530 AD to 1066 AD. The η Aquarid and Comet Halley orbital elements reported in the literature are reproduced in Table 7.4 at the end of the chapter (page 242).

The mean orbit for stream meteors detected by the AMOR radar should resemble the orbit of the progenitor comet. Using the AMOR radar to observe the η Aquarid shower therefore provides an astronomical check on the performance of both the radar and reduction package. This represents the final operational test of the radar. Two distinct requirements must be met.

It is first necessary to identify those meteors that are members of the η Aquarid stream. All the shower meteors will come from the same region of the sky. Only those with a heliocentric velocity in the appropriate bounds

are included. In simple terms the AMOR radar measures where a meteor was coming from and how fast it was going. Essentially the two criteria, radiant and speed are used to identify stream members. From these stream meteors a mean orbit can be estimated. This is done by inspecting the orbital elements of those meteors that come from the region of the radiant. For comparison a serial search using the modified Drummond D-Criteria measure of orbital similarity is used as an alternative method for identifying stream members. This gives a similar set of shower meteors but selected directly from their calculated orbital elements.

Successfully observed and reduced meteor events are the product of a complicated selection process. The geometry of the radar and the influence of the Earth conspire to produce overall orbital distributions that reflect the way the data was acquired and reduced rather than the *in situ* state. It is necessary to ensure that the mean stream orbit produced is not the result of observational selection effects and the way in which stream members were identified. That is, the final orbit must be shown to be a meteor stream of astronomical origin. Table 7.4 includes a summary of the results from the chapter.

7.2 Identifying Stream Meteors

7.2.1 Radiant Position

Meteor streams are named after a bright star in the region of sky from where the meteors appear to radiate on the day of peak activity. This apparent radiant point can also be defined by right ascension and declination. The coordinates for the η Aquarids shower (Lindblad, 1989) and the star η-Aquarius (Norton's Star Atlas) from which it takes its name are

	R.A.	Dec	
η-Aquarius	338.°3	-0.°4	star
η Aquarids	337.°7	-1.°6	shower

$L_{\odot\ 1950} = 45.°8$

Whilst the apparent radiant point is convenient for identifying the origin of visual meteor showers it has less direct relevance in calculating radar meteor orbits. The correction for zenith attraction depends on how close to transit the shower is when the meteor is observed. In fact the reduction package does not calculate the *true* apparent radiant point but rather the corrected geocentric radiant. This is the radiant after the atmospheric deceleration, Earth's rotation and zenith attraction have been allowed for. It is not until this point that the kinematic information is converted to equatorial coordinates. The difference between the apparent radiant and the corrected geocentric radiant is small. It can be as much as 10° but typically is less than 2°. for η Aquarid meteors this correction will usually be less than 0.5°. The corrected geocentric

radiant is used throughout this chapter. The radiants are given for the mean epoch of date.

Shower activity for the η Aquarid shower extends from about April 28 to May 12. The corrected geocentric radiants of all meteors observed in the period 1990 April 28 to May 18 are plotted in Figure 7.1. Only the area of interest $320° < \alpha < 360°$ and $-9° < \delta < +4°$ is shown. A clumping of observations is evident around the radiant point. The enhancement of meteors coming from this part of the sky is most obvious in the distribution of long duration overdense meteor trails. These correspond to larger shower meteors in general. The *clumping* of meteor observations around the radiant is the most direct evidence for shower activity.

As the Earth moves through the stream the apparent radiant will move. Meteoroid orbits are calculated from observations made as the particle crosses the plane of the ecliptic. In the case of η Aquarid shower meteors this will be at the descending node. As the Earth moves through the stream the ascending node of the detected meteors must also change. If the other orbital elements remain constant the shift in ascending node causes the radiant point to move more or less parallel to the ecliptic. Cook (1972) gives a daily motion of $+0.9°$ in right ascension and $+0.4°$ in declination. Lindblad (1989) also describes the daily motion in right ascension and declination for his mean orbit as follows.

$$\alpha = 0.888(L_\odot - 45.°79) + 337.38 \quad (7.1)$$

$$\delta = 0.456(L_\odot - 45.°79) - 2.°01 \quad (7.2)$$

The solar longitude, $L_\odot$, is given in the 1950 epoch. The daily shift in the radiant for Lindblad's mean orbit is overlaid on Figure 7.2(a)[1]. The η Aquarid *clump* of meteor radiants is stretched in this direction. This feature is important in indicating an astronomical significance for the grouping.

As a first approximation to selecting shower meteors, right ascension and declination boundaries were placed around the radiants with an enhanced distribution; $330° < \alpha < 350°$ and $-5° < \delta < 3°$. This box had the advantage that selecting those observations within the limits is very easy to set up. A better scheme takes into account the daily motion of the radiant. This is a rectangle with long sides parallel to the motion described in equations (7.1) and (7.2). The actual position of the bounding box is chosen by eye. All those observations with radiants within the box drawn on Figure 7.2 are identified as candidate shower meteors. This gives 520 candidate observations from a total of 1283 in the right ascension declination region plotted.

[1]This line has been added as the astronomers say *to guide the eye.*

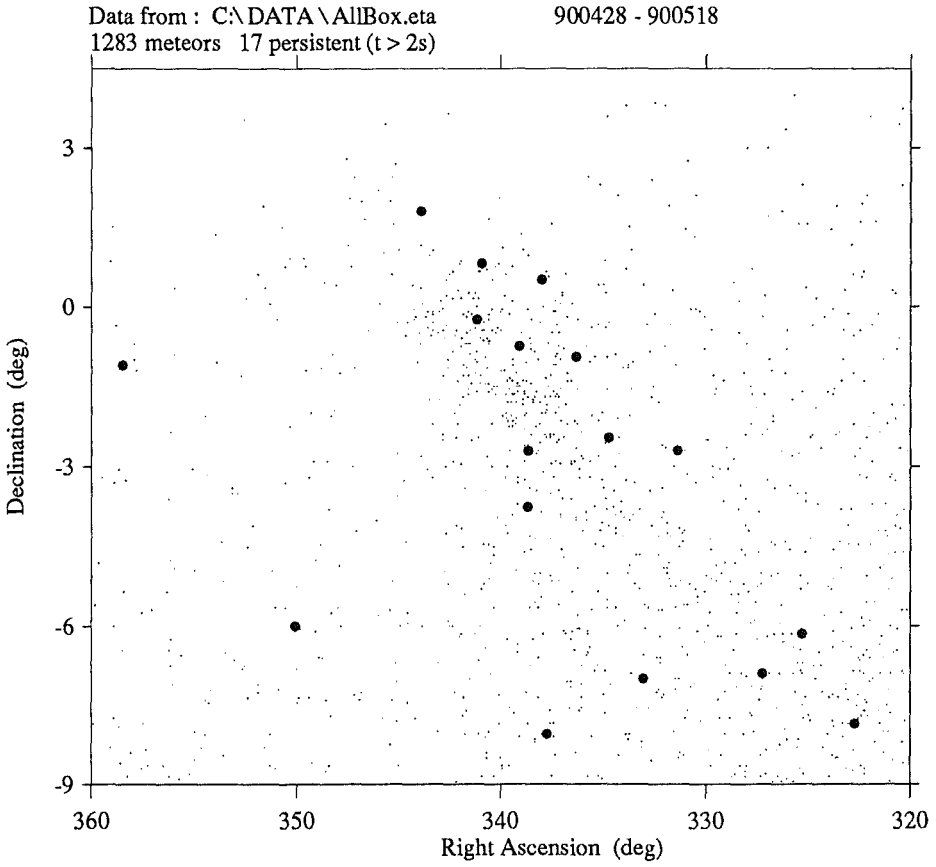


Figure 7.1: Corrected geocentric radiant points for meteors with radiant points in the region of the η Aquarid stream. All data in the region from the period 1990 April 28 to May 18 are shown. The larger dots are persistent meteor trails that lasted longer than two seconds representing meteors with a radar magnitude of +5 or brighter.

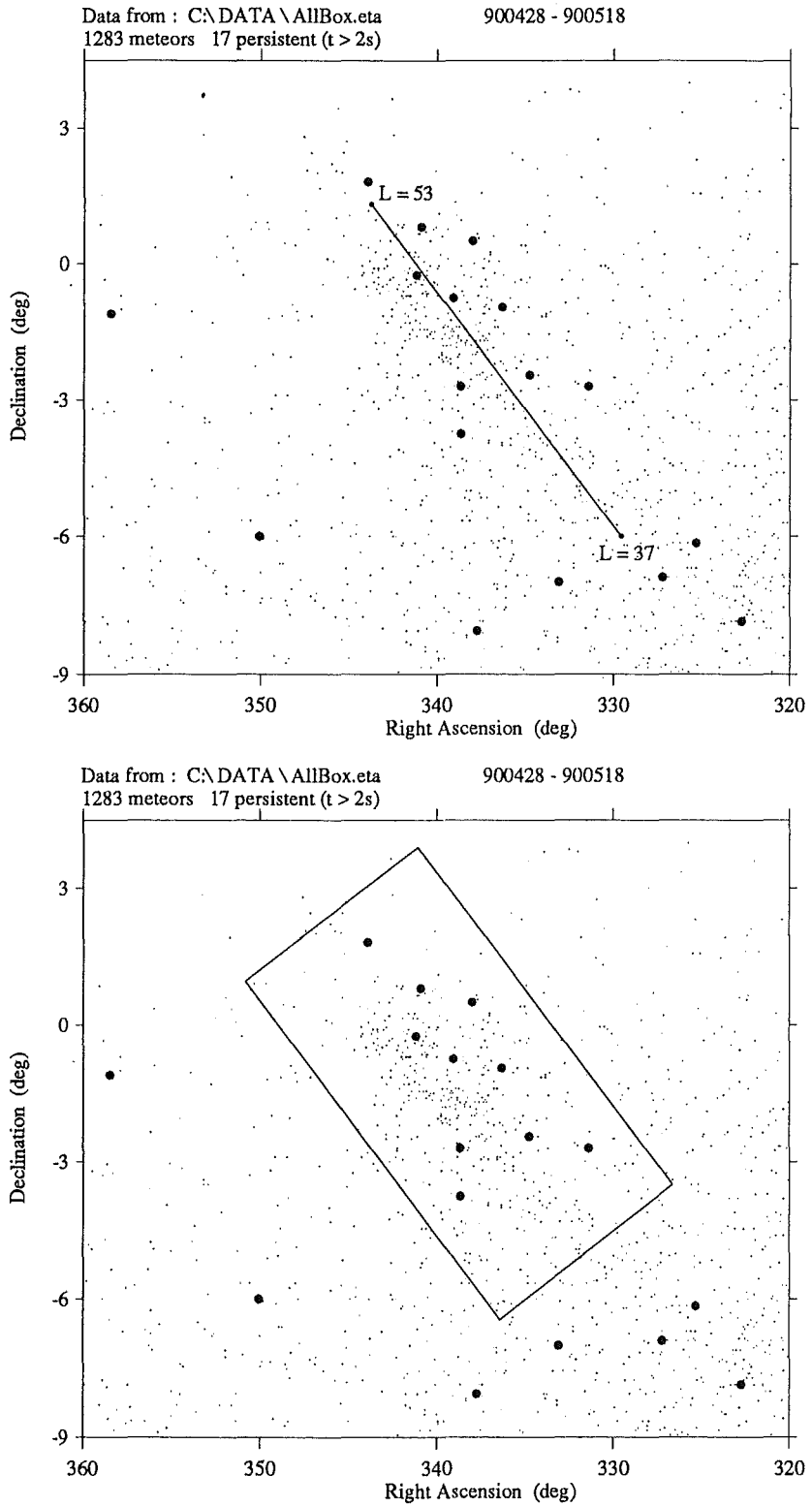


Figure 7.2: (a) The daily motion of the radiant point for the mean orbit of the stream is overlaid on the data from Figure 7.1 (Lindblad, 1989). It runs from April 28 ($L_\odot = 37$) to May 15 ($L_\odot = 53$). (b) The radiant box used to identify candidate shower meteors is overlaid. It is aligned parallel to the daily motion.

7.2.2 Heliocentric Velocity

Meteors which come from the same part of the sky and have similar velocities will travel in similar orbits. A concentration of particles moving in the same orbit constitutes a meteoroid stream. How fast the meteoroid was moving at any point in the orbit fixes the orbit size. Defining a particular range of velocities provides a reasonable criterion for stream membership. These bounds on velocity should eliminate sporadic meteors contained within the radiant box found in the preceding section.

Largely because it showed a more peaked distribution I chose to impose this velocity membership criterion by means of the heliocentric velocity. The distribution of heliocentric velocities for all the meteors in Figure 7.1 is shown in Figure 7.3(a). The bounds were set by inspecting the velocity distribution of those meteors within the radiant box, Figure 7.3(b).

The heliocentric meteoroid velocity must fall in the range $34 < v_m < 48\text{km.s}^{-1}$. This final membership criterion gives a total of 361 η Aquarid meteors detected during 1990. The final heliocentric velocity distribution for the stream is displayed for comparison in Figure 7.3(c).

7.2.3 Membership Criteria

The radiant and velocity membership criteria for the η Aquarid shower can be expressed mathematically. The right ascension, α, declination, δ, and heliocentric velocity, v_m are used to do this.

$$1.95\,\delta + 333.^\circ5 < \alpha < 1.95\,\delta + 349^\circ \quad (7.3)$$

$$-3.315\,\delta + 315.^\circ0 < \alpha < -3.315\,\delta + 354^\circ \quad (7.4)$$

$$34\text{km.s}^{-1} < v_m < 48\text{km.s}^{-1} \quad (7.5)$$

The radiants of all the η Aquarid meteors are plotted in Figure 7.4(a). The sporadic (non-shower) meteor radiants from the whole region are displayed in Figure 7.4(b) for comparison. To give a complete picture of the observed data the observed velocity distribution for the shower meteors is plotted in Figure 7.5. A number of issues are raised by this selection of stream members. They are discussed in detail below. The majority of persistent meteors associated with the radiant are not included as members of the shower. The distribution of sporadic meteors implied by the selection needs to be understood.

Persistent non shower members

The distribution of sporadic meteors still includes a distinctive concentration of overdense (persistent trails lasting longer than 2 seconds) in the immediate region of the shower radiant. Stream membership for those north of -5.5° declination is rejected based on their individual heliocentric velocities. These

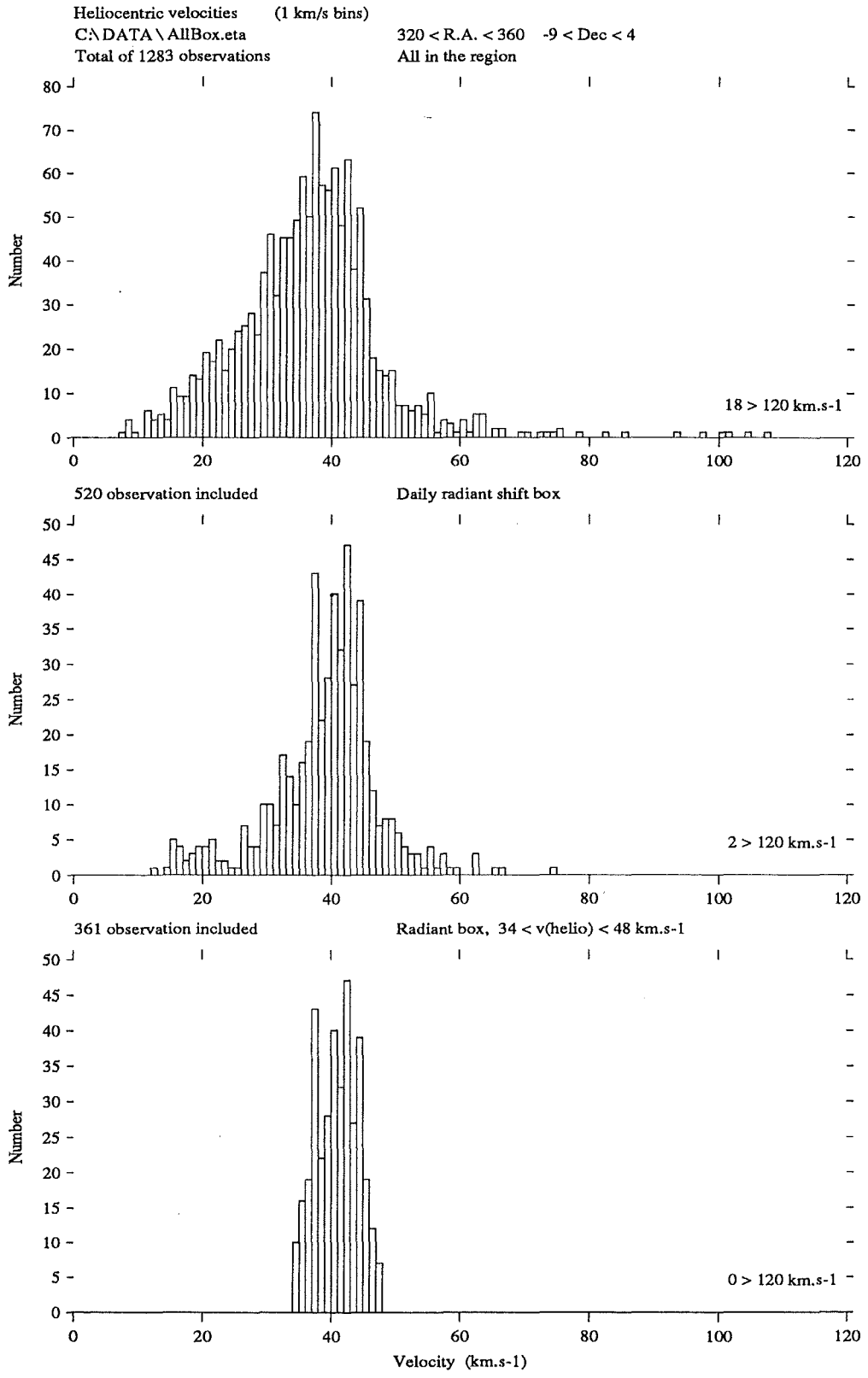


Figure 7.3: The heliocentric velocity distribution for three different right ascension declination limits are plotted. (a) All the observations plotted on Figure 7.1; $320° < \alpha < 360°$, $-9° < \delta < +4°$. (b) The radiant box from Figure 7.2. (c) Stream members with the heliocentric velocity criteria imposed, $34 < v_m < 48 \text{km.s}^{-1}$.

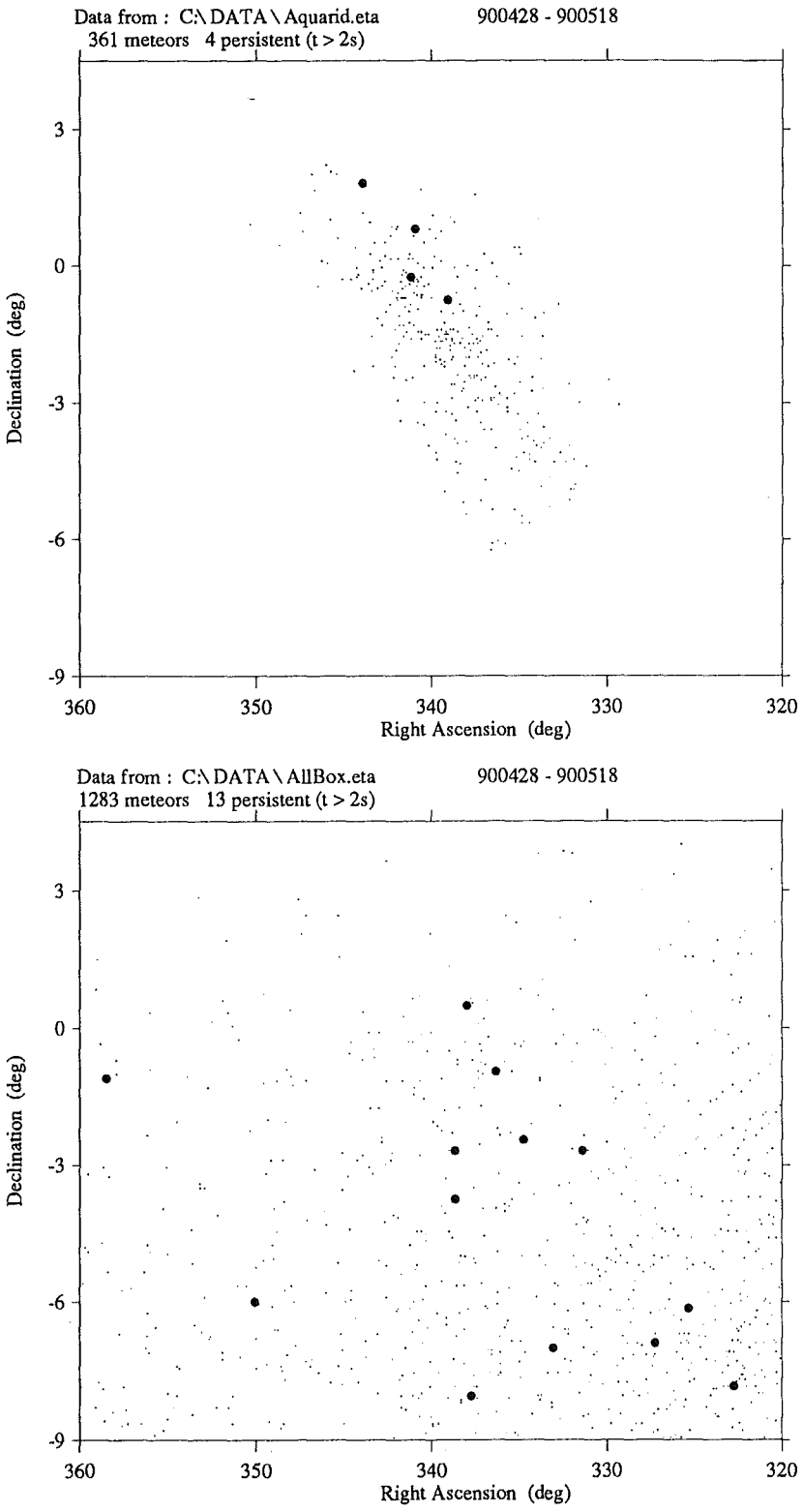


Figure 7.4: The radiants of 361 stream members (a) are compared with the 922 rejected as being part of the sporadic complex (b).

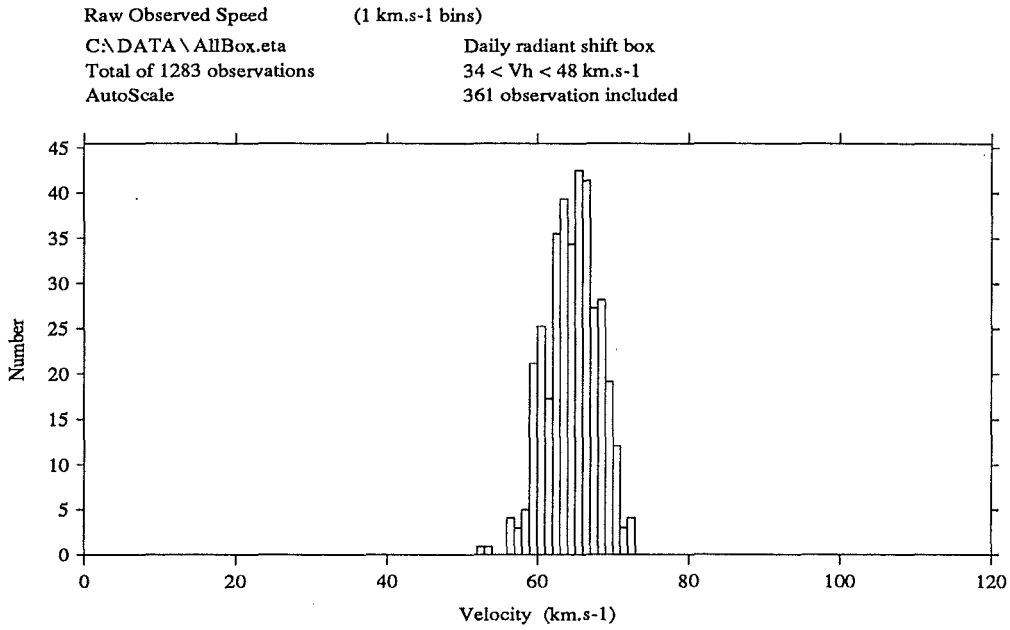


Figure 7.5: The observed velocity distribution for shower meteors selected using the radiant-velocity criteria.

ranged from 26.7km.s^{-1} to 52.5km.s^{-1}. It is surprising that only four of the ten overdense trails that fall within the η Aquarid radiant box are included in the rather loose heliocentric velocity stream membership criterion. The presence of so many long duration meteors concentrated near the radiant is a strong indicator of shower activity (see Figure 7.1).

The presence of these non shower persistents cannot be simply explained by assuming timing errors in the velocity determinations. The reduction package uses the ratio of the time lags between two pairs of receiver stations to estimate the radiant point of the meteor trajectory. Any error in these timelag estimates would have to be in an identical ratio for both receiver pairs.

The observing software identifies the continuation of a persistent trail as being any echo at a similar range and within 2 seconds of a previous detection. Where a second meteor closely follows another, the initial meteor will be mistakenly identified as a persistent. Meteors forming part of a shower tend to arrive at similar ranges[2]. The meteor rate during a shower is also higher. The likelihood of detecting these misidentified trails is thus increased.

However since there are many more shower than non shower meteors in the radiant box there should be a predominance of misidentified persistent meteors with η Aquarid orbits. In fact there are more long duration non shower

[2]The range of shower meteors shifts slowly as the shower radiant moves across the sky during the day. The feature is used by meteor rate radars to try and separate the shower component from the sporadic rate.

meteors. It seems unlikely that the over abundance of non shower persistent meteors near the η Aquarid radiant can be explained by misidentified persistent trails. Additionally the persistent meteors plotted on Figure 7.4 all last longer than 2 seconds. The observing code would require at least two additional redetections for persistent trails of this duration.

Larger meteoroids are more subject to fragmentation. For any fragmentation event the AMOR system will detect the first particle to reach a specular reflection point. This could be a slower fragment displaced at a slight angle and therefore detected sooner. Alternatively a small piece might shatter and move more rapidly along the meteor trajectory. In both these cases the radiant would be largely unchanged but the observed velocity would be affected. Other fragments from the original meteoroid would contribute to the ionisation in the trail increasing its duration. Shower meteoroids tend to be larger than the sporadic complex increasing the probability of fragmentation occurring.

Sporadic distribution near radiant

The non shower component in the region of the η Aquarid radiant is plotted in Figure 7.4. The variations in this background sporadic distribution need to be discussed. Aside from the residual clumping of persistent echoes discussed in the previous section the density of sporadic radiants varies fairly smoothly across the region. With increasing right ascension the number of meteor detections decreases. Moving north from -9° to +3° declination the number of detections also decreases.

Figure 7.6 shows the radiant distribution for all meteors detected on one day during the shower. The radiants are concentrated between -30° and 0° declination. The radar beam is about 30° wide in elevation and as the Earth rotates it sweeps out an arc in this declination range. The region of interest for the η Aquarid shower is right on the edge of the radar beam. Moving out to the edge of the beam will be accompanied by a reduction in the radar sensitivity. Fewer meteors will be detected as the distance from the beam centre is increased.

During the η Aquarid shower the earth continues to rotate around its orbit. The position of the meridian for 0700hrs on April 28 and May 12 is plotted on Figure 7.6. This corresponds to a 14° shift in right ascension. Fewer observations at large right ascensions will be obtained on the earlier days in the shower.

Mostly observations continued through to around 0900hrs. Generally by that time sea echoes off sporadic-E ionisation had started to significantly interfere with and mask meteor observations. The effective end to observations ranged from shortly after 0700hrs through to 1000hrs. This atmospheric effect tended to decrease the number of meteor observations at larger right ascensions.

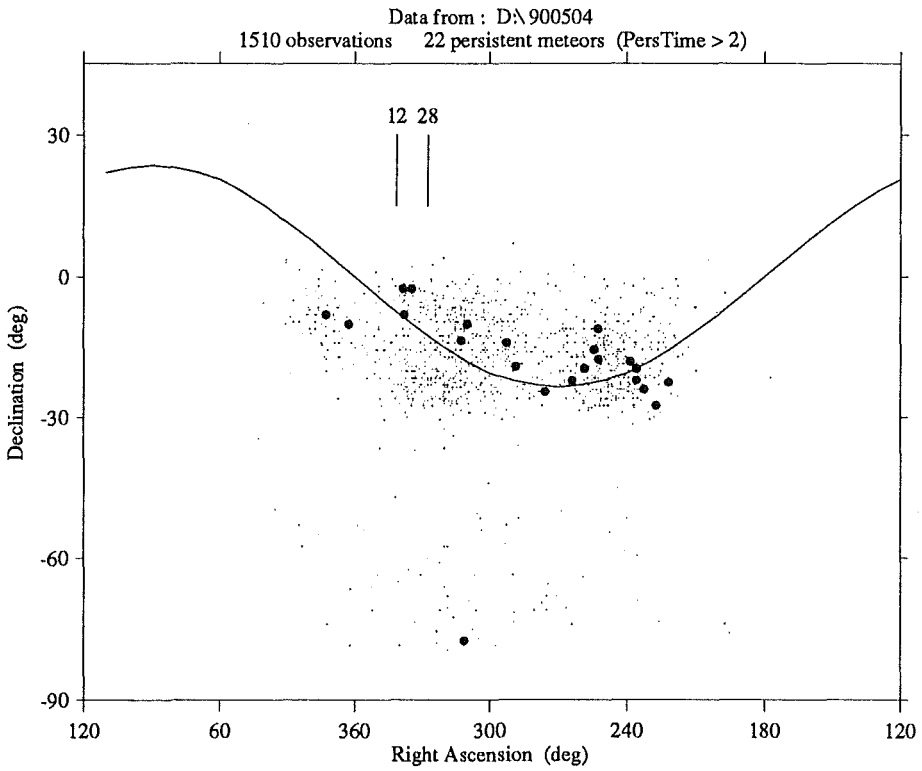


Figure 7.6: The corrected geocentric radiants for all meteors detected on 1990 May 4. The curved line shows the position of the ecliptic. The two vertical lines show the position of the meridian at 0700hrs on April 28 and May 12.

The distribution of sporadic meteoroids is concentrated in the plane of the ecliptic. Where the radar is aligned to detect radiants from the ecliptic the meteor rate will be enhanced. The position of the ecliptic is overlaid on Figure 7.6. Where the position of the ecliptic on the meridian reaches its most southerly limit ($\alpha = 270°$) there is a noticeable dearth of meteors. For right ascensions where the ecliptic falls more directly in the radar beam (between -20° and -10°) the rate is enhanced. As the right ascension increases from 320° the ecliptic moves north from -10° declination. The number of meteors detected drops as the ecliptic moves out of the beam.

The general distribution of sporadic meteors in the region of the radiant is relatively smooth. It can be explained by characteristics of the Earth's motion, equipment parameters and the concentration of meteoroid particles near the ecliptic. There does not seem to be any localised distortion by the observing system that could deceptively produce a *shower* concentration. Against this smooth variation the η Aquarid shower meteors represent a distinct concentration. By implication the shower detected by the AMOR system should be of astronomical significance.

7.3 Daily Motion of the Radiant

The radiant points for the η Aquarid shower meteors are plotted in Figure 7.7. The average radiant positions for meteors observed on each day are overlaid on the diagram. As the Earth moves through the shower these average points tend toward larger right ascensions and more northerly declinations. This is exactly the motion expected for particles in similar orbits detected at slightly different ascending nodes. The daily motion of the radiant is the first such demonstration of this effect with orbits determined from radar meteors. It provides another confirmation that the radiant grouping is of astronomical origin.

The radiant criterion for selecting shower meteors was based on an assumed daily motion of the radiant. The 361 observations selected as η Aquarid shower meteors do show a daily motion in the average position of their radiants. This is along the same line as that suggested by Lindblad (1989) and discussed earlier. The selection of shower meteors based on the radiant box aligned parallel to the ecliptic is vindicated.

7.4 Distribution of η Aquarid Orbital Elements

The distributions of the orbital elements for the 361 η Aquarid stream meteors are plotted in Figures 7.8 and 7.9. The selection criteria based on radiant position and heliocentric velocity, described in section 7.2, was used to identify the shower meteors.

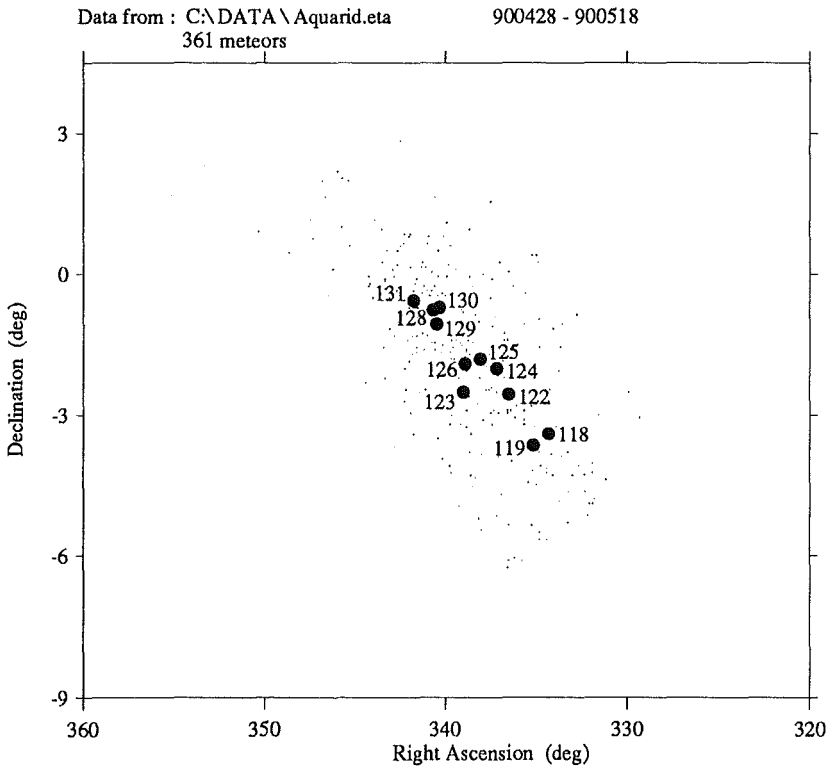


Figure 7.7: Average radiant points for the η Aquarid shower meteors detected on each day during the shower. These are identified by day number which increases as the Earth moves through the shower.

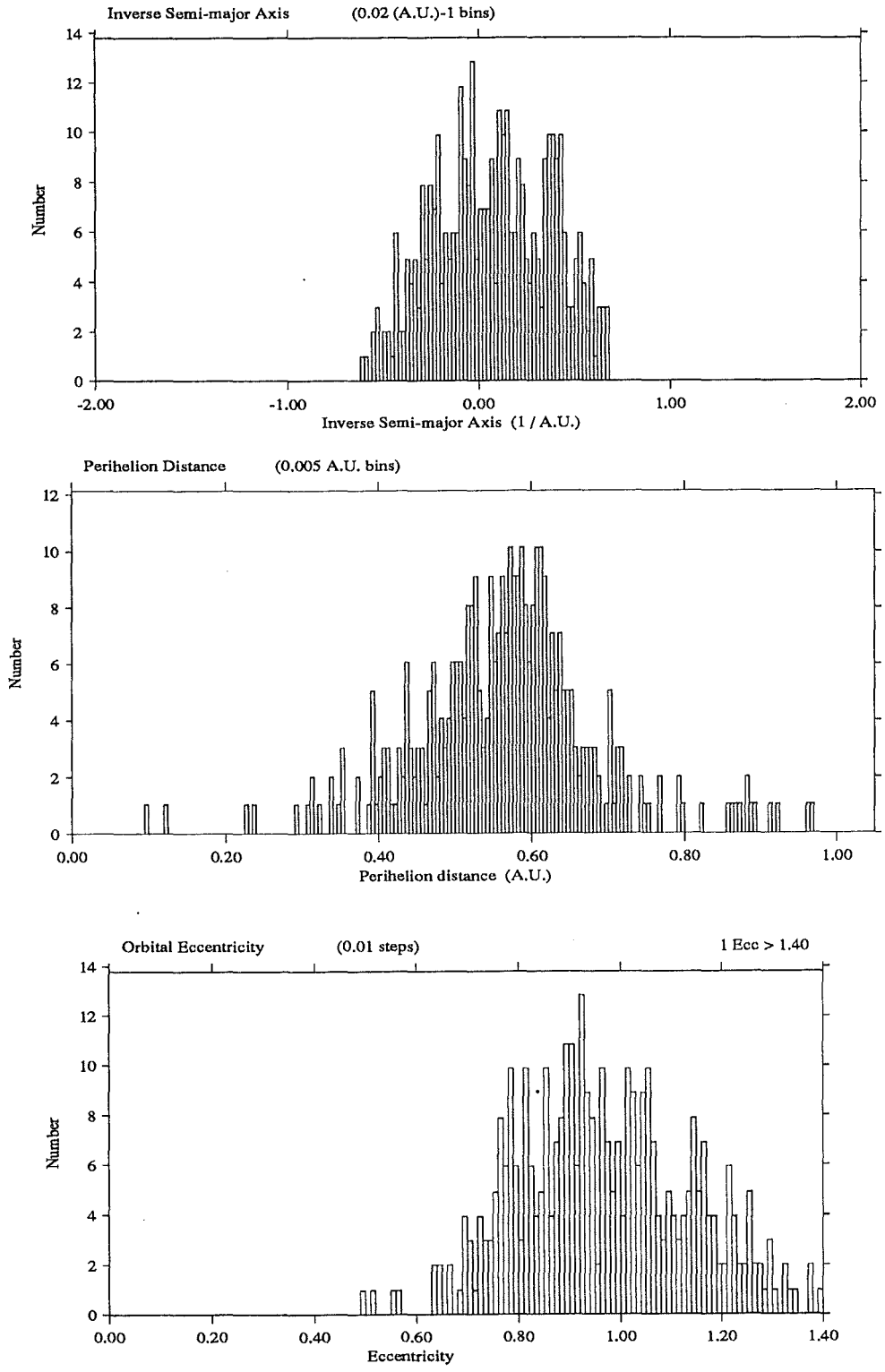


Figure 7.8: The orbital element distributions for 361 η Aquarid shower meteors selected using the radiant box and heliocentric velocity criteria. Stream values are:

(a)	Inverse semi-major axis	a^{-1}	=	0.0351 A.U.$^{-1}$
(b)	Perihelion distance	q	=	0.57 A.U.
(c)	Eccentricity	e	=	0.98

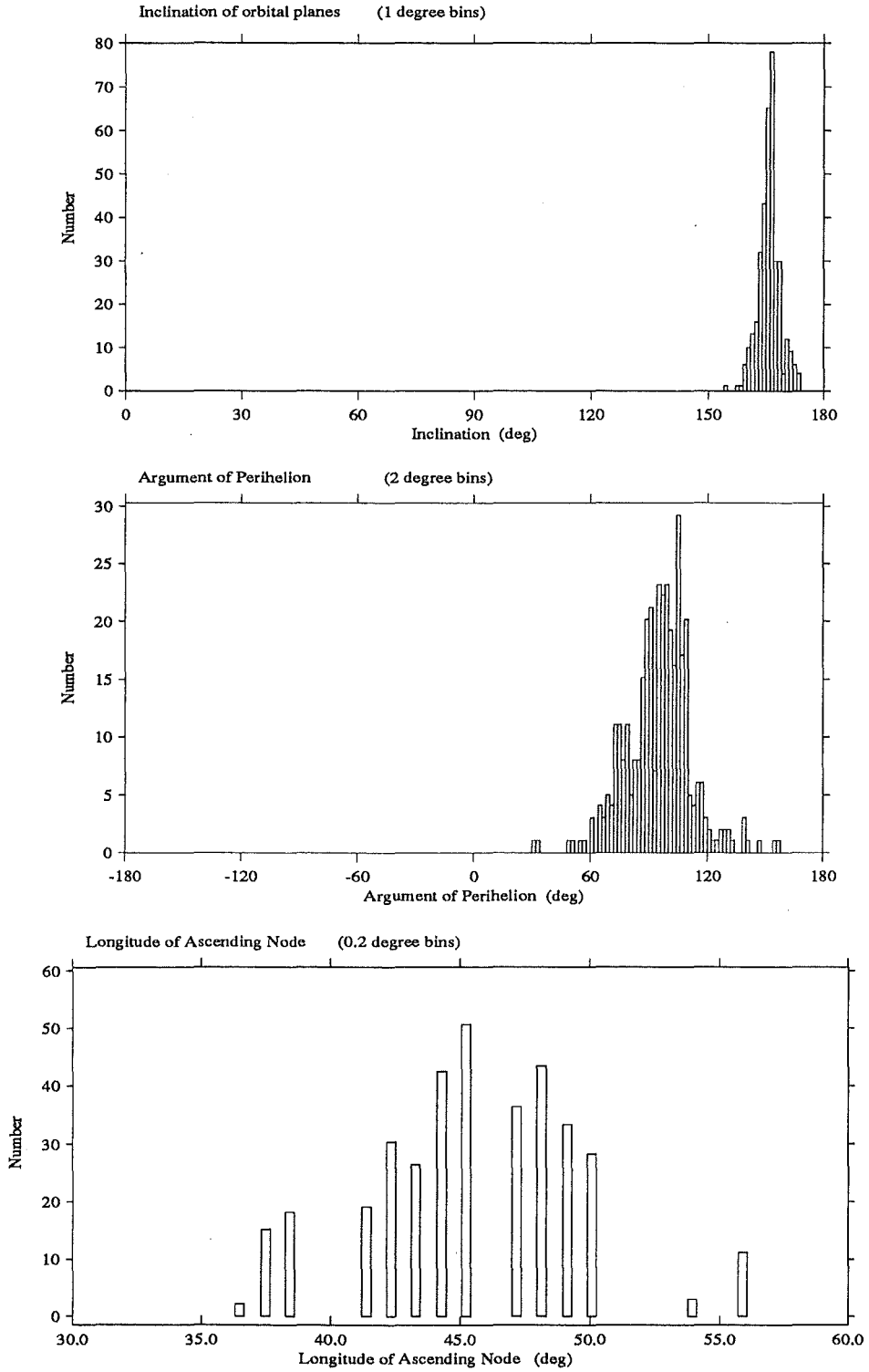


Figure 7.9: The orbital element distributions for 361 η Aquarid shower meteors selected using the radiant box and heliocentric velocity criteria. Stream values are:

(a)	Inclination	i	=	165.5°
(b)	Argument of Perihelion	w	=	97°
(c)	Longitude of Ascending Node	Ω	=	46°

The semi-major axis is related to the kinetic energy of the orbit. By comparing with the average orbital velocity of the Earth, $v_\oplus$, the inverse semi-major axis for a meteoroid detected at 1 AU can be determined mathematically.

$$a^{-1} = 2 - \frac{v_m^2}{v_\oplus^2} \tag{7.6}$$

The heliocentric velocity limits on membership fix the cutoffs in the inverse semi-major axis plot. The limits $34 < v_m < 48$ km.s^{-1} correspond to inverse semi-major axis limits of $0.70 > a^{-1} > -0.60$ AU^{-1}. These cutoff limits are also reflected in the distribution of orbital eccentricity.

The distribution of orbital inclination remains the most distinctive feature of the shower. The perihelion distance and the argument of perihelion exhibit a clumped distribution around the expected η Aquarid stream values. The shower meteors are all detected at the descending node of their orbit. The longitude of the ascending node is therefore the same as the longitude of the Sun at the time of observation. The maximum of shower activity occurs at around $L_\odot(1950) = 46°$. This corresponds to transit on May 7 for the Christchurch radar. The longitude for the ascending node is dependent on the length of time and on which days the radar was operational. The value chosen is close to the mean for shower meteors detected by this radar. It is consistent with the peak in shower activity observed using meteor rate radars (McIntosh and Hajduk, 1983). The mean longitudes agree with other estimates from meteor orbits, Lindblad (1989), see Table 7.4.

Orbital elements for the η Aquarid stream are tabulated in Table 7.4. Average values for the orbital elements of the shower meteors detected by the AMOR system have been calculated. The standard deviation for the respective distributions about these means are also included. The standard error of the mean is the standard deviation divided by the square root of the number of observation included in the mean. The values are repeated here.

Radiant-Velocity selection, Radiant box, $34 < V_m < 48$ km.s^{-1}

	q	e	i	w	Ω_{1950}	V_m	Number
	0.566	0.969	165.8	95.4	45.8	41.0	361
s.e.	0.006	0.009	0.1	0.9	0.2	0.2	
s.d.	0.118	0.171	2.8	16.7	4.1	3.3	

Using a mean value assumes that the distribution of the orbital elements is reasonably symmetric. Inspection of Figures 7.8 and 7.9 clearly indicates this is not the case. The peaks in these distributions have been chosen to define the η Aquarid stream orbit. These values are tabulated in Table 7.4 as the Stream Orbit, AMOR Christchurch 1990. These values are confirmed in section 7.6 using a D-Criteria stability search.

Stream Orbit, AMOR Christchurch 1990.

q	e	i	w	Ω_{1950}	V_m	Number
0.57	0.98	165.5	97	46	41.5	

7.4.1 Measurement Uncertainties

Before further analysis the orbital distribution obtained the measurement uncertainties need to be discussed. The radar observations for each meteor are reduced to a set of station data. This includes the time of the observation, the elevation angle of the echo and time-lags associated with the passage of the meteor through the specular reflection point at the three receiver sites. The size of the uncertainties associated with these measurements are discussed in the chapter on station data reduction. The uncertainties are noted in Table 7.2. These are an estimated one standard deviation error.

Implicit in the calculation of the orbit is the assumption of an infinitely narrow radar beam directed south. In fact this beam is a few degrees wide. The implications of this uncertainty are calculated based on a one degree error in the azimuthal angle of the velocity vector. The range from the radar station to the meteor echo is accurate to 6 km. The position of the heliocentric velocity of the meteor is assumed to be that of the earth at the time of observation so uncertainties in the range have no bearing on the accuracy of the orbital elements. The time of observation is read from a hardware clock in the computer, typically set and accurate to within one second. Imposing a one minute error in the time of observation illustrates how much of an error is necessary before any slight uncertainties can be discerned in the accuracy of the orbital elements. The following discussion assumes a typical η Aquarid meteor observation recorded at 1990 May 7, 0735:00hrs. The values of the measured parameters give orbital elements close to those adopted for the stream orbit. The combination was lifted from a particular meteor observation with the appropriate elements rather than calculated!

Table 7.2 in essence tabulates the propagation errors in the orbital elements due to measurement uncertainties. The orbital elements for the station data used as an empirical base are tabulated at the top of the table. Each of the station parameters are varied from the base value by the one standard deviation uncertainty while the other data is held constant. The orbital elements are then calculated to illustrate the error due to that measurement uncertainty.

The uncertainties need to be compared with the distributions of Figures 7.8 and 7.9. The maximum possible range in each orbital element, given the worst possible combinations of the errors in the station data is tabulated in Table 7.3. The station data associated with the η Aquarid base orbit is noted at the top of Table 7.3.

The range in orbital elements implied by the measurement uncertainties

	q	e	i	w	a^{-1}	V_{helio}
η Aquarid stream test orbit 1990 May 7						
	0.57	0.98	165.2	96.9	0.04	41.6
Lag12, SE-NW timelag, $t_{12} = 11.4 \pm 1$ pulse						
10.4	0.51	1.02	166.6	92	-0.03	42.3
12.4	0.62	0.94	163.8	102	1.04	40.8
Lag23, N-S timelag, $t_{23} = -15.5 \pm 1$ pulse						
-14.5	0.68	1.23	165.1	115	-0.34	45.4
-16.5	0.45	0.85	165.2	78	0.34	38.2
Elevation, $\psi = 42.2° \pm \frac{1}{2}$ degree						
41.7	0.56	0.94	166.0	95	0.10	40.8
42.7	0.58	1.02	164.4	99	-0.03	42.3
Azimuth, $A = 0 \pm 1$ degree						
-1	0.52	0.92	165.2	89	0.16	40.1
+1	0.61	1.06	165.2	104	-0.10	42.9
Time, 0735:00, absolute worst $\pm$ 1 minute						
0734	0.58	0.98	165.0	97	-0.10	41.5
0736	0.56	0.98	165.4	96	-0.10	41.6

Table 7.2: The propagation errors associated with the uncertainties in the observed station data. Each of the data values are varied by plus and minus the uncertainty and the corresponding orbital elements are tabulated. In each case all the other station elements are held constant. The orbital elements associated with the test orbit are included at the top of the table.

η Aquarid stream test orbit 1990 May 7

Observation		0735:00		± 1 minute
Time lags	t_{12}	= 11.4		± 1 pulse
	t_{23}	= -15.5		± 1 pulse
	t_{13}	= $t_{12} + t_{23}$	=	$t_1 - t_3$
Elevation	ψ	= 42.2		± $\frac{1}{2}$ degree
Azimuth	A	= 0.0		± 1 degree

	t_{12}	t_{23}	ψ	A	DD value
Perihelion distance 0.57 AU					
0.33	10.4	-16.5	41.7	-1.0	0.27
0.77	12.4	-14.5	42.7	1.0	0.15
Eccentricity 0.98					
0.76	12.4	-16.5	41.7	-1.0	0.13
1.45	10.4	-14.5	42.7	1.0	0.19
Inclination 165.5°					
163.0	12.4	-14.5	42.7	-1.0	0.02
167.5	10.4	-16.5	41.7	1.0	0.02
Argument of perihelion 97°					
62	10.4	-16.5	41.7	-1.0	0.19
126	12.4	-14.5	42.7	1.0	0.17
Heliocentric speed 41.6 km.s^{-1}					
35.6	12.4	-16.5	41.7	-1.0	0.13
48.5	10.4	-14.5	42.7	1.0	0.19
Inverse semi-major axis 0.04 AU^{-1}					
0.55	12.4	-16.5	41.7	-1.0	0.13
-0.67	10.4	-14.5	42.7	1.0	0.19

Table 7.3: The uncertainties in the elements for the η Aquarid stream orbit are estimated. The station data are varied within the expected measurement error and the corresponding variation in the orbital elements tabulated. In each case the combination of station data is chosen to give the most extreme difference in the element concerned. The implication for the Drummond D-criteria when this extreme value is compared with the stream element has also been evaluated (see section 7.5.

closely reflects the cutoffs in the distributions. The broad range in eccentricities can be explained by measurement uncertainties. The sharp peak in the orbital inclination is consistent with experimental errors. The η Aquarid stream members detected by the AMOR system and selected using the radiant box and heliocentric velocity could all come from the same orbit. Any structure in the individual distributions of the orbital elements can be explained as emerging from the measurement uncertainties.

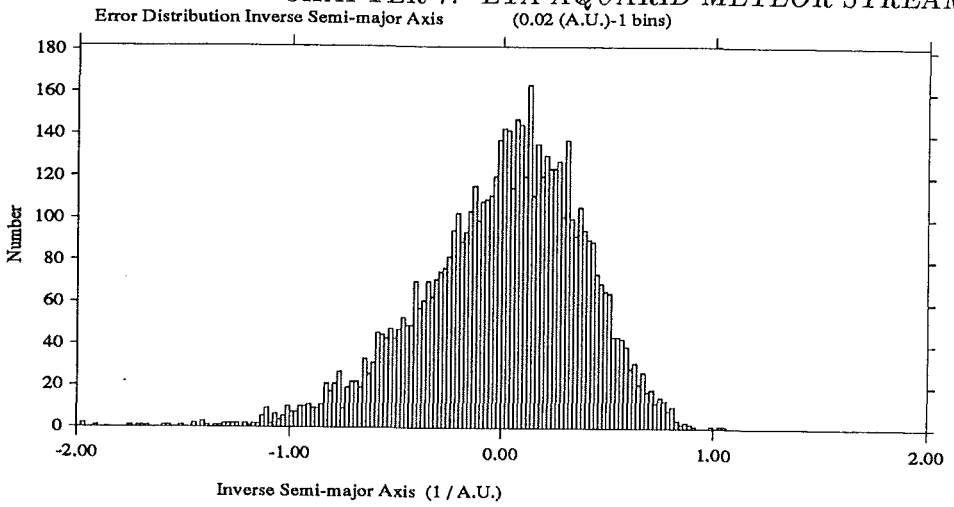
The uncertainty data from Table 7.3 gives error limits to the orbital elements for any individual member of the stream. A picture of the implied distributions due to uncertainties in the station data is necessary to decide whether uncertainties in the station data explain the orbital element distributions. Each of the station parameters, two time-lags, elevation and azimuth angles, were randomly varied within a gaussian distribution centred on the base orbit used earlier in the section. The orbital elements were then calculated. A series of 6000 samples were taken and the orbital element uncertainty distributions plotted in figures 7.10 and 7.11.

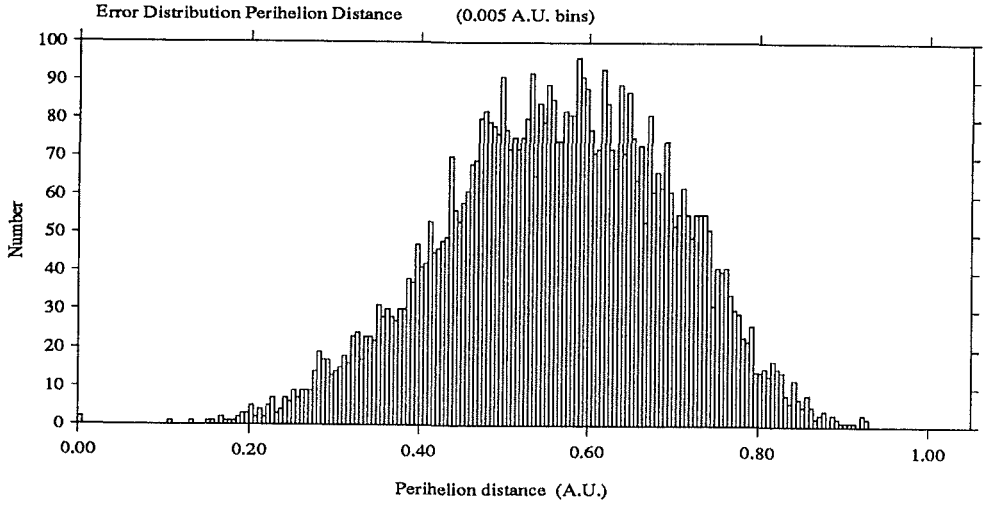
The distribution of orbital elements due to measurement uncertainties closely mirrors that of the η Aquarid stream meteors in Figures 7.8 and 7.9. All the random trial orbits are calculated for an observation made around transit at the peak of shower activity, 1990 May 7, 0735:00hrs. The sample therefore has no variation in the longitude of the ascending node. Aside from noting the daily shift in the radiant point no significant effort has been made to investigate how the orbital elements change over the 15 days of the shower. An understanding of this is essential to extend the error analysis to variations in longitude of the ascending node. The close similarity of the uncertainty distributions to the actual observed orbital element distributions make extraction of information about the underlying stream structure very difficult.

The distribution of heliocentric velocity in the bottom panel of Figure 7.11 indicates the accuracy possible in velocity determinations. Some heliocentric velocities up to around 50 km.s^{-1} are the result of measurement uncertainties for what should be bound orbits. This has implications in the search for meteors that were in true hyperbolic orbits. It does indicate that velocities above 60 km.s^{-1} are unlikely to come from simple measurement errors in the time-lags.

7.4.2 Background Element Distributions

To provide a backdrop against which to compare the shower results this section presents the overall element distributions for all orbits observed during the η Aquarid observations. The discussion of the sporadic distribution near the radiant in Section 7.2.3 does this for the meteor radiants in the immediate region of the shower. During the period of the 1990 η Aquarid shower a total of 18132 meteor observations were made that could be reduced to heliocentric orbits. The number of orbits obtained on each day is plotted as a function





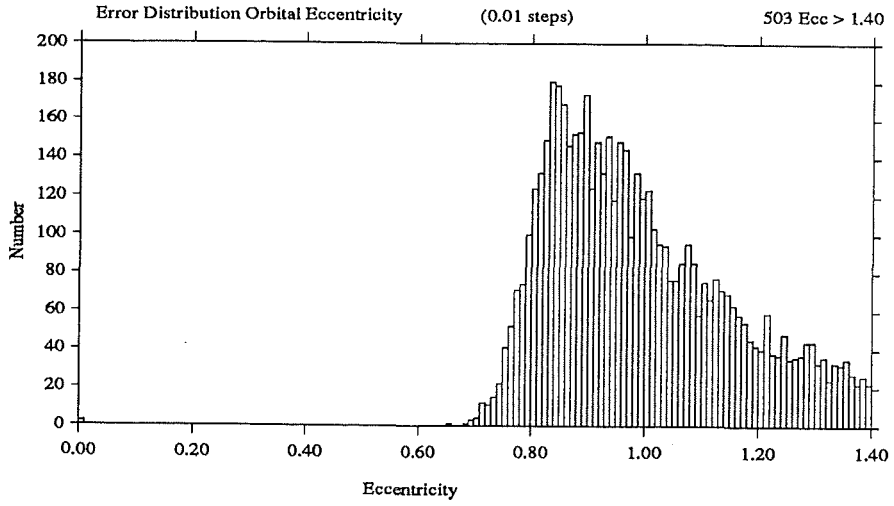


Figure 7.10: The orbital element distributions for measurement uncertainties imposed on an η Aquarid test orbit. A sample of 6000 random variations of the station data within a normal distribution centred on the values associated with the test orbit. The one sigma uncertainties are tabulated in Table 7.2.

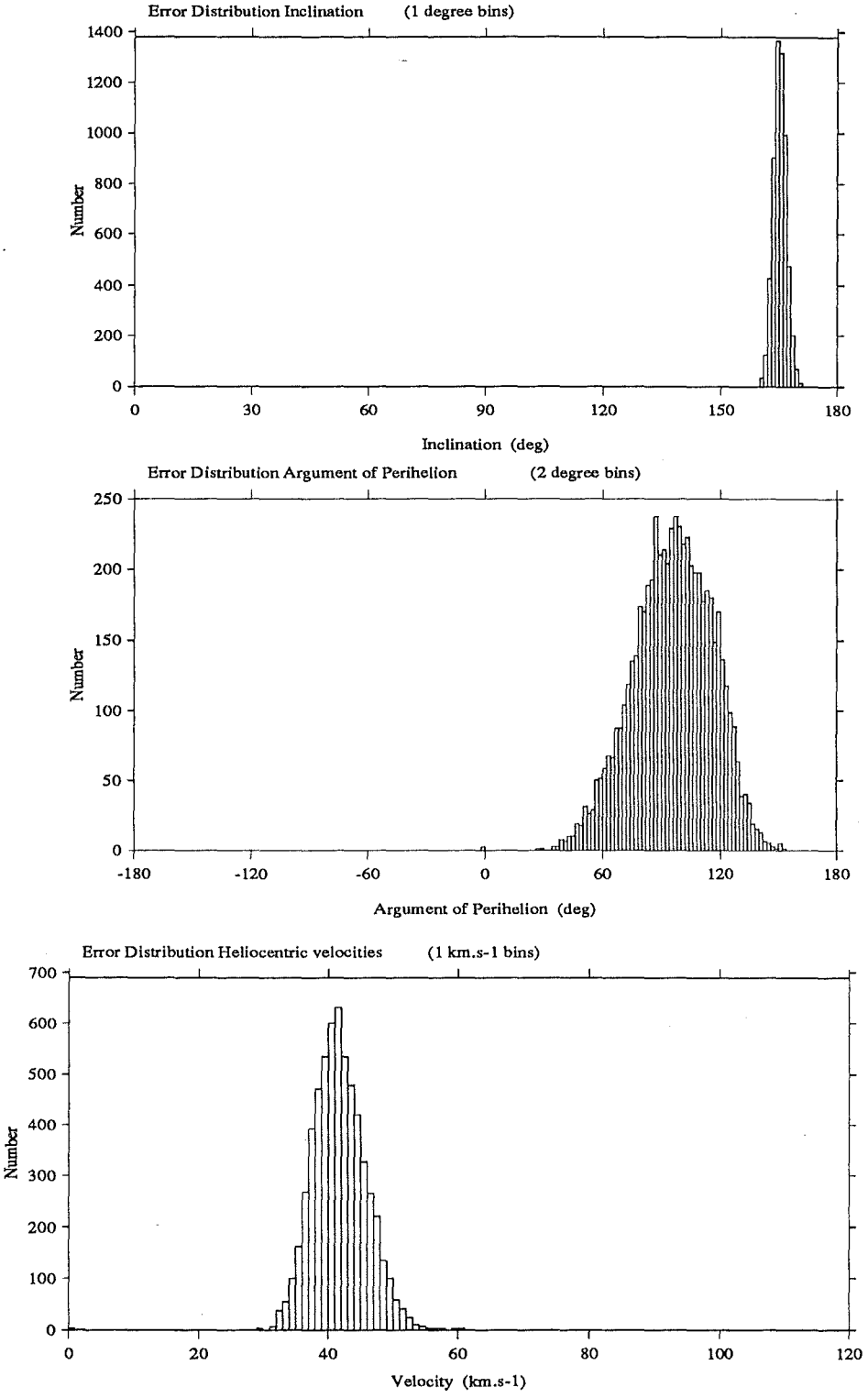


Figure 7.11: The orbital element distributions for measurement uncertainties imposed on an η Aquarid test orbit. A sample of 6000 random variations of the station data within a normal distribution centred on the values associated with the test orbit. The one sigma uncertainties are tabulated in Table 7.2.

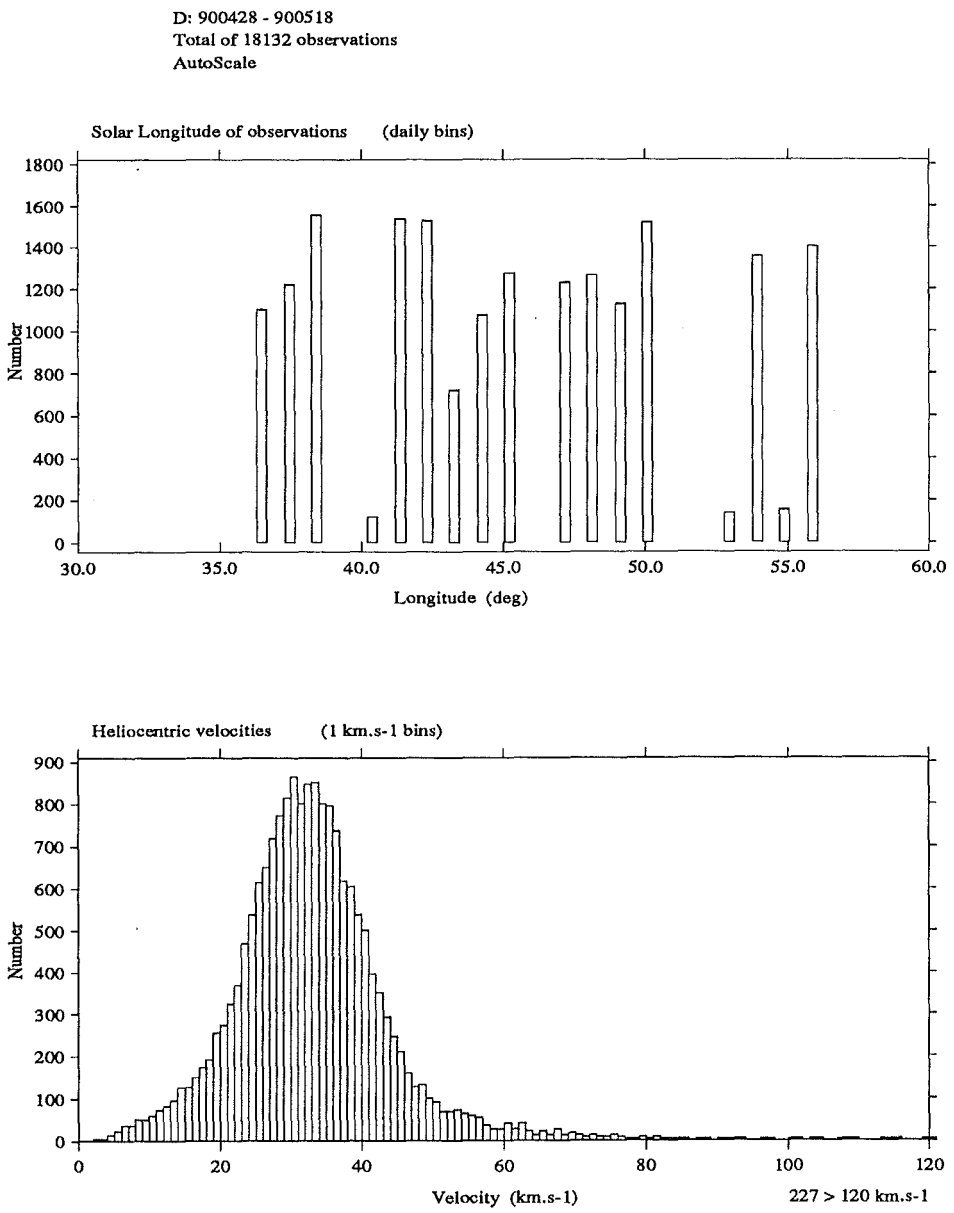


Figure 7.12: The number of observations on each day are plotted as the solar longitude at the time of observation. The heliocentric velocity distribution for all 18132 meteor orbits observed in the period 1991 April 28 to May 18 is plotted in the bottom panel.

of the solar longitude in Figure 7.12. The heliocentric velocity for all meteors is included in the bottom panel as a comparison for the velocity membership criterion.

The orbital element distributions for all the data collected during the shower are plotted in Figures 7.13 and 7.14. Except for a slight enhancement at the stream inclination there is no evidence of the stream in these total distributions.

A similar set of distributions can be produced for the non shower meteors in the region of the radiant. See Figure 7.4 for the radiant distribution of this sample. These very closely resemble those for the total data set. Almost all of this sporadic component in the region of the shower are detected after perihelion. The argument of perihelion ranges fairly uniformly between 0° and 180° but includes hardly any with negative values.

7.4.3 Sporadics with Stream Inclinations

The one distribution that does not closely follow the overall distributions is that of the inclination. This shows a significant excess in the number of meteors with inclinations between 163° and 172°. The orbital inclinations for non shower meteors in the region of the η Aquarid radiant is plotted in Figure 7.15. Conservation of angular momentum and the presence of the Sun in the plane of the orbit make the inclination of a particle the most difficult element to change with typical interplanetary perturbation forces. A number of meteors might be expected where the only remaining similarity with the original orbit is the inclination of the orbital plane. During the evolution of the particle the size of the orbit would also be expected to shrink. A scatter plot of inclination against heliocentric velocity was composed to see whether this excess was associated with particles whose orbital energy had been reduced by interplanetary perturbations. Figure 7.16 plots all the meteors from the region of the shower. The larger dots are those identified as shower meteors. Amongst these the orbital inclination becomes less defined at smaller heliocentric velocities. A similar concentration is only faintly discernible amongst non-shower orbits where strongly perturbed stream members might lie.

Shower meteors with smaller velocities seem to occupy retrograde orbits more steeply inclined to the ecliptic. Slower stream particles are likely to have spent longer evolving away from the comets orbit than meteoroids in the stream core. The probability of collisions with the general sporadic complex is more likely for orbits more closely inclined to the ecliptic. Collisions will tend to preferentially remove particles with inclinations closer to 180°.

The inclination of the orbit for η Aquarid meteors is only slightly affected by measurement uncertainties, see Table 7.3. Any problems in identifying the location of the maximum rising slope in the echo profile could render the other orbital elements wildly divergent whilst still giving a reasonable inclination[3].

[3]The whole thing might just be an artifact of experimental uncertainties !

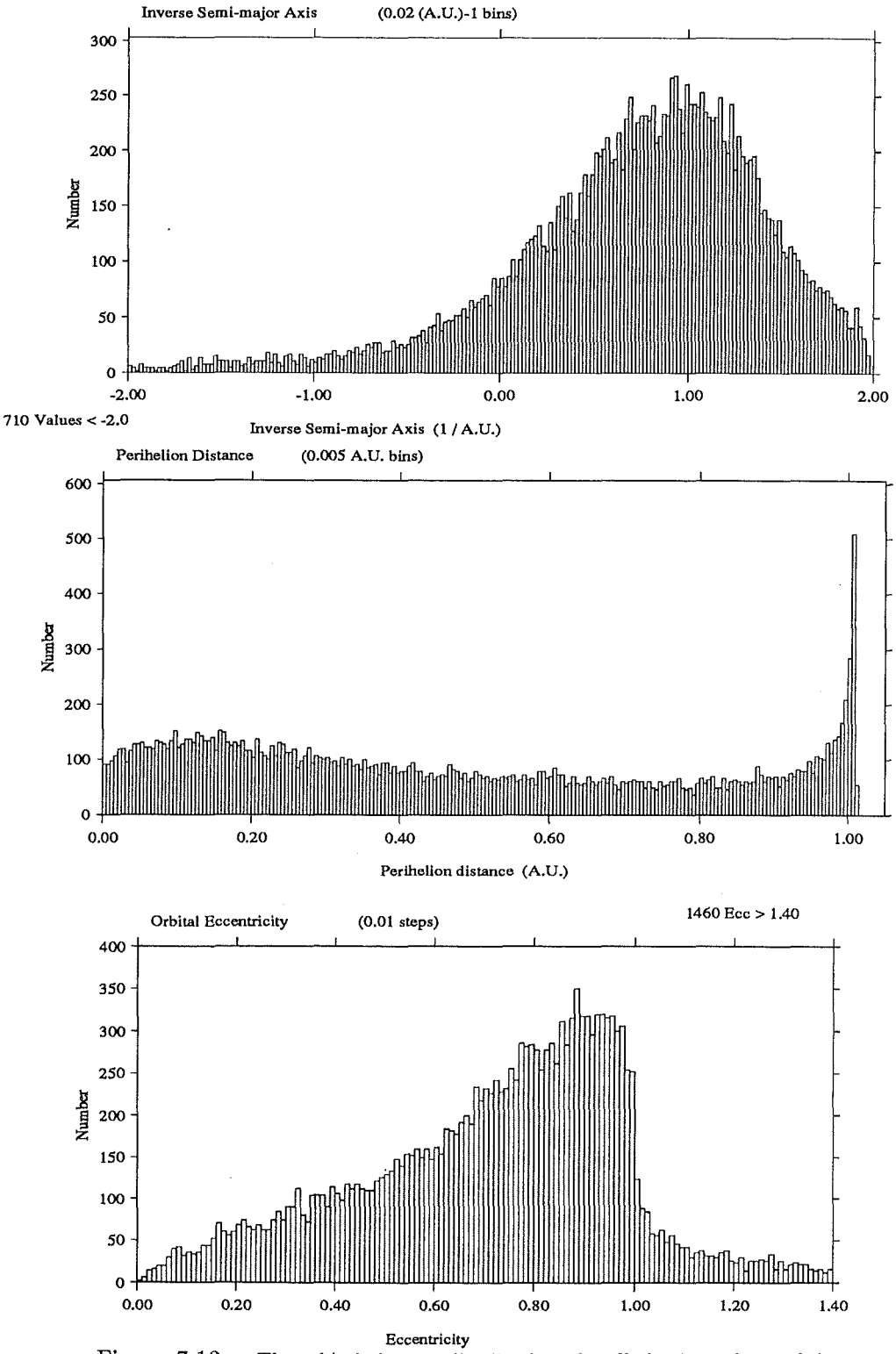


Figure 7.13: The orbital element distributions for all the data observed during the period of the η Aquarid shower. A total of 18132 observations were made in the period 1990 April 28 to May 18.

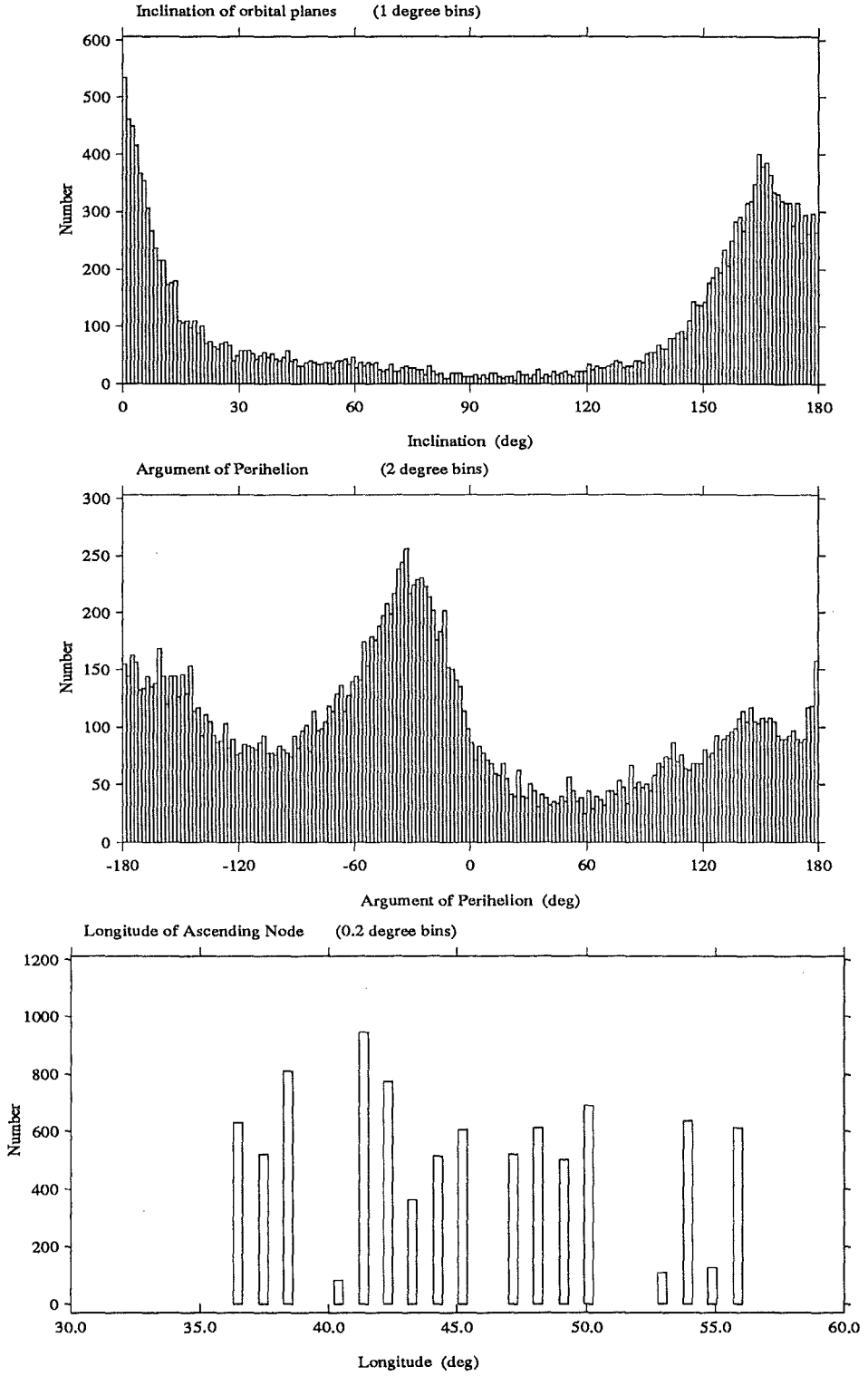


Figure 7.14: The orbital element distributions for all the data observed during the period of the η Aquarid shower. The third panel shows the longitude of the ascending node. These represent the 8963 meteors detected at their descending node.

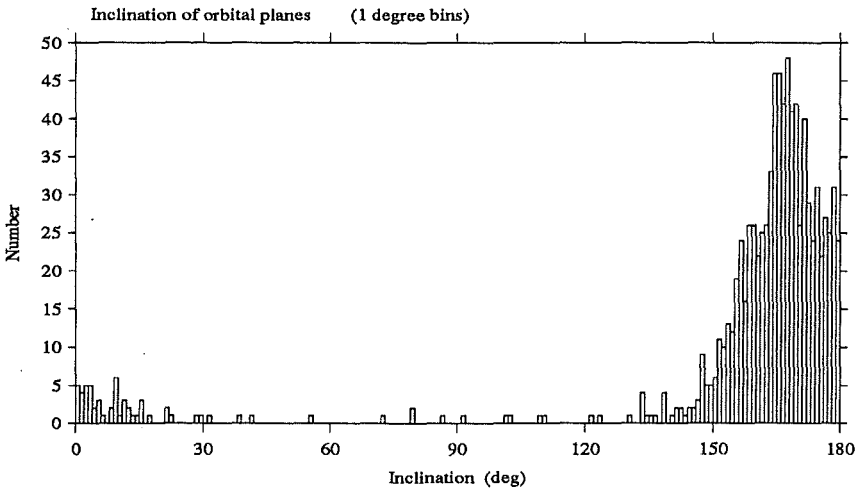


Figure 7.15: The inclination for the 922 non shower meteors with radiants in the right ascension, declination region plotted in Figure 7.4. An excess residue of observations with inclinations typical of the η Aquarid stream can be seen between 163° and 172°.

This excess of inclinations amongst the non shower meteors from the region of the η Aquarid radiant may be related to the *sporadic* persistent meteors in the same region noted in Section 7.2.3. The rising edge of persistent echoes are often less well defined. This would provide an answer to the sporadic inclination excess that is consistent with the concentration of persistent non shower meteor trails.

7.5 Drummond D-Criteria

The η Aquarid meteors discussed thus far in this chapter have been identified by their radiant and velocity. It is possible to identify shower meteors directly from their calculated orbital elements. Inspection of the elements for meteors from the region of the η Aquarids shows a preponderance of orbits with inclinations, 160° to 170° and perihelion distances 0.5 to 0.7 A.U. Where one or more elements are distinct from the majority of sporadics, as in this case, identification of the shower is relatively straight forward. More generally all five orbital elements that define a meteoroid orbit need to be compared. Each orbit occupies a point in the corresponding five dimensional parameter space. The presence of a stream can then be discerned where there is a significant enhancement in the number of meteor orbits around a common point in this space.

Looking for structure in a five dimensional phase space is conceptually difficult. To make comparisons between orbits more manageable a measure

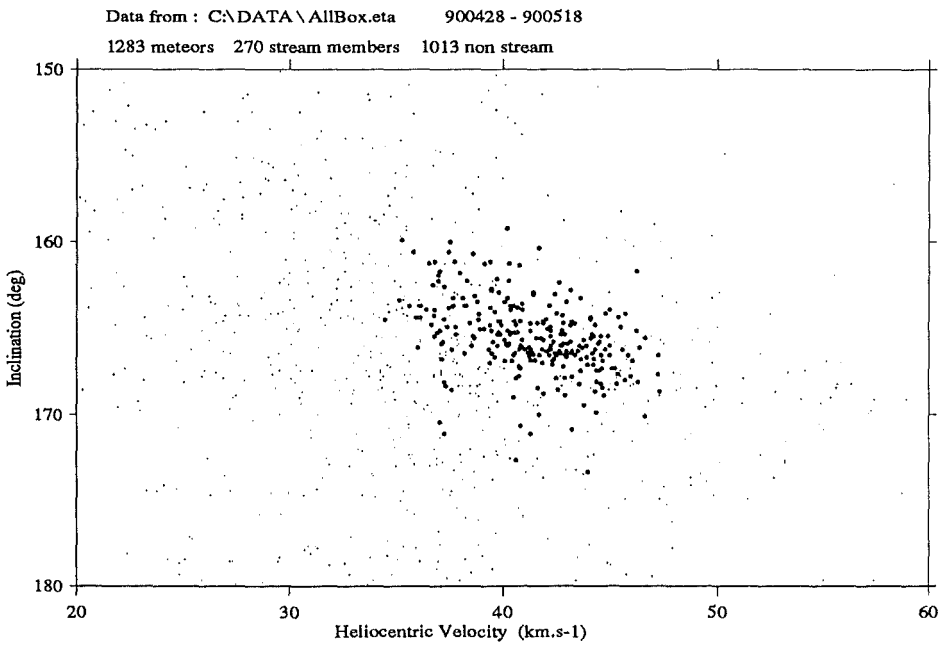


Figure 7.16: Orbital inclination and heliocentric velocity at detection are plotted against each other. Only observations from the region of the shower radiant are included, Figure 7.4. Shower meteors selected using the D-Criteria serial search are plotted as larger spots.

of similarity between two orbits is needed. Southworth and Hawkins (1963) introduced a quantitative measure of orbital similarity. This D-Criterion was used as a discriminant to identify meteors belonging to major showers and define new minor streams. They constructed a four dimensional curvilinear coordinate system based on the differences between the orbital elements for the two orbits to be compared. The Southworth-Hawkins discriminant provides a scalar measure for the *distance* between two orbits. Drummond (1981) refined this geometric measure of similarity to render each of the terms dimensionless. He also adjusted the weighting to ensure each term varied linearly from zero to one. The D-Criteria as modified by Drummond is used in the subsequent sections of this chapter. The actual form of the discriminant is discussed here.

The coordinates used to make the comparison are q, e, I and Θ, where I is the angle between the orbital planes and Θ is the difference between the longitudes of perihelion measured from the intersection of the orbits. The ecliptic latitude, β and longitude, λ are used to calculate Θ. For a pair of orbits labeled 1 and 2 the Drummond D-Criteria, DD, is defined mathematically.

$$\mathrm{DD} = \left(\frac{e_1 - e_2}{e_1 + e_2}\right)^2 + \left(\frac{q_1 - q_2}{q_1 + q_2}\right)^2 + \left(\frac{I}{180°}\right)^2 + \left(\frac{e_1 + e_2}{2}\frac{\Theta}{180°}\right)^2 \quad (7.7)$$

$$I = \cos^{-1}\left[\cos i_1 \cos i_2 + \sin i_1 \sin i_2 \cos\left(\Omega_1 - \Omega_2\right)\right] \quad (7.8)$$

$$\Theta = \cos^{-1}\left[\sin\beta_1 \sin\beta_2 + \cos\beta_1 \cos\beta_2 \cos\left(\lambda_1 - \lambda_2\right)\right] \quad (7.9)$$

$$\beta = \sin^{-1}\left(\sin i \sin w\right) \quad (7.10)$$

$$\lambda = \Omega + \tan^{-1}\left(\cos i \tan w\right), \quad \text{add } 180°, \text{ if } \cos w < 0 \quad (7.11)$$

7.5.1 Selecting Stream Members

Two alternative methods can be employed for selecting stream members using the Drummond D-Criteria. Stream members can be identified directly by meeting a D-Criteria level of similarity with a mean orbit. An appropriate mean stream orbit is chosen and any meteor orbit within the given D-Criteria range is said to be a shower meteor. The orbit can be iterated to provide the largest number of shower meteors. The second possible method of selecting stream members is to require each of them to be closely associated with another stream member. This serial search allows a form of *chaining* to link stream orbits without being biased by the form of the D-Criteria function. In this sense the serial search is superior, though clearly it does require more computing time.

It remains to choose a significance level for the D-Criterion. A number of workers propose different values; Drummond (1979) suggests values between 0.10 and 0.175 while Olsson-Steel (1987) uses 0.125. Lindblad (1971) in applying the Southworth-Hawkins D-Criteria to precise photographic orbits concluded the limit should be $0.80N^{-\frac{1}{4}}$, where N is the number of meteors in the sample. For the 18132 orbit observations made during the 1990 η

Aquarid shower this represents a limit of 0.07. The corresponding Drummond D-Criteria level would be around 0.04.

The actual value chosen will depend on the number of meteor orbits in the data set near the stream concerned. Visual meteor orbits can be calculated to far greater precision than radar observations and so more stringent values of the discriminant are justified. The lower limits are more appropriate to the serial search technique. The direct search limits should take more account of the spread in orbital elements due to measurement uncertainties. The implication for D-Criteria calculations of the errors in typical η Aquarid elements are tabulated in Table 7.3. Having used a number of different levels to identify η Aquarid meteors in a direct D-Criteria search there seems little *a priori* difference between them.

The η Aquarid orbit identified in Section 7.4 is used to provide a base orbit for the direct D-Criteria membership search. Figure 7.17 identifies the stream meteors based on two different D-Criteria values. In both cases the radiants agree well with those selected in Section 7.2. The weaker criterion, $DD < 0.15$ identifies less than 10% which are not included in the radiant-velocity membership criteria. The distribution of orbital elements based on this D-Criteria selection are similar to those obtained in Section 7.4. The shape of the orbital distributions for members selected by a direct D-Criteria search is always biassed by the expression used to estimate the similarity with the stream orbit, equation (7.7). You tend to get exactly the distribution of orbital elements you expect!

A D-Criteria serial stream search was conducted with membership requiring $DD < 0.04$ with at least two other stream members[4]. The first generation of members was seeded using the mean orbit from Section 7.4. The radiant points of the stream members are plotted in Figure 7.18. In comparison with the direct search far fewer outlying meteors are included. Only four fall just outside the radiant box velocity membership criteria.

7.5.2 D-Criteria Element Distributions

The orbital element distribution for stream members selected using the D-Criteria serial search are plotted in Figures 7.19 and 7.20. These 270 orbits provide a more tightly defined group of shower meteors. They will exhibit less sporadic contamination but probably reject more diffusely associated orbits. Both sets of membership criteria provide a similar picture of the distribution of orbital elements for the shower. The mean orbital elements calculated from the D-Criteria members are repeated in Table 7.4 for easier comparison.

[4]Christened sponsors or parents in the programs concerned.

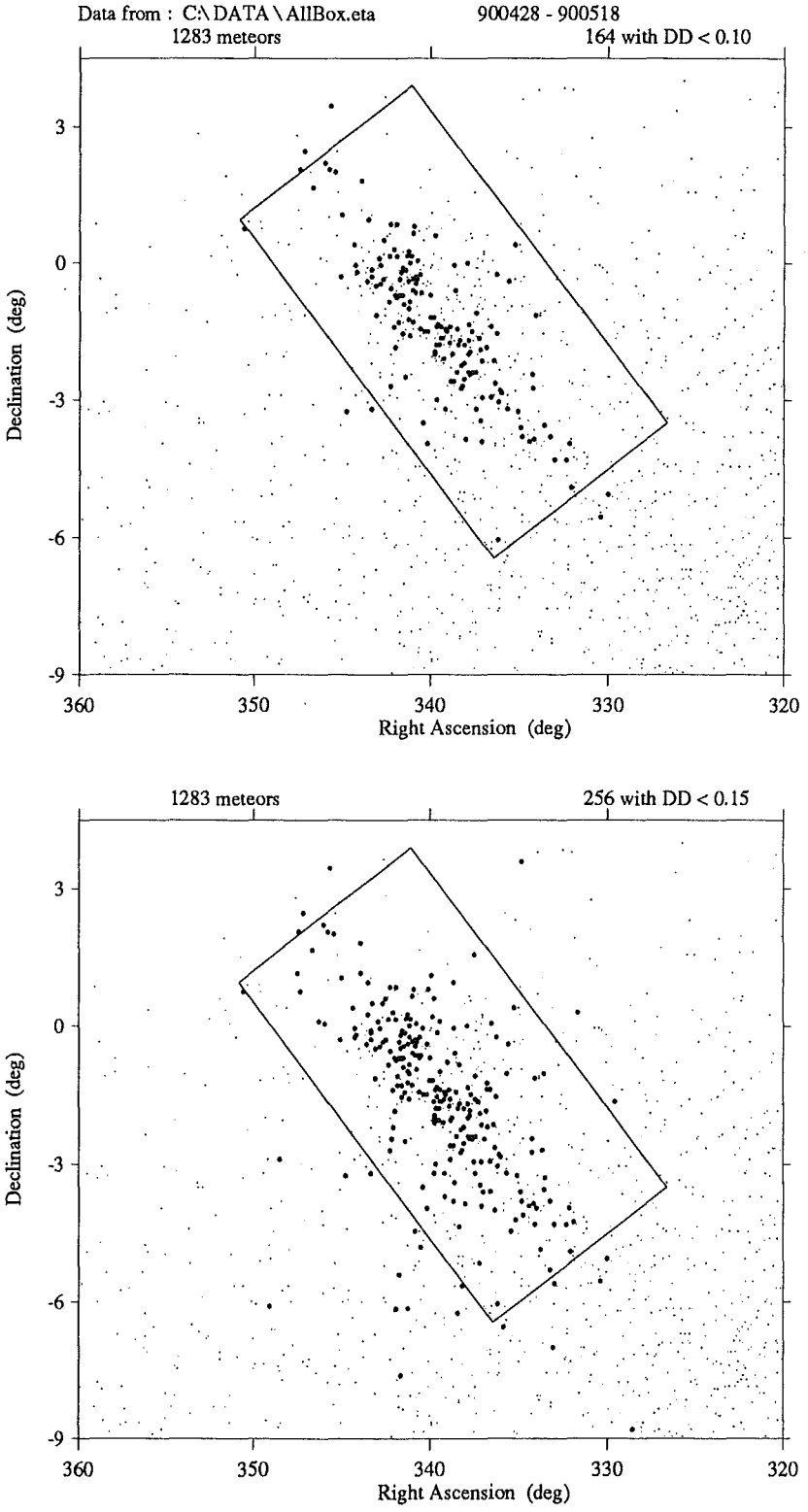


Figure 7.17: The radiants of stream members identified using the Drummond D-Criteria are plotted as larger spots. The stream orbit from Section 7.4 is used to provide the comparison. (a) 164 Meteors with DD < 0.10 of which 8 are outside the radiant box. (b) 25 of 256 are outside the radiant box with DD < 0.15.

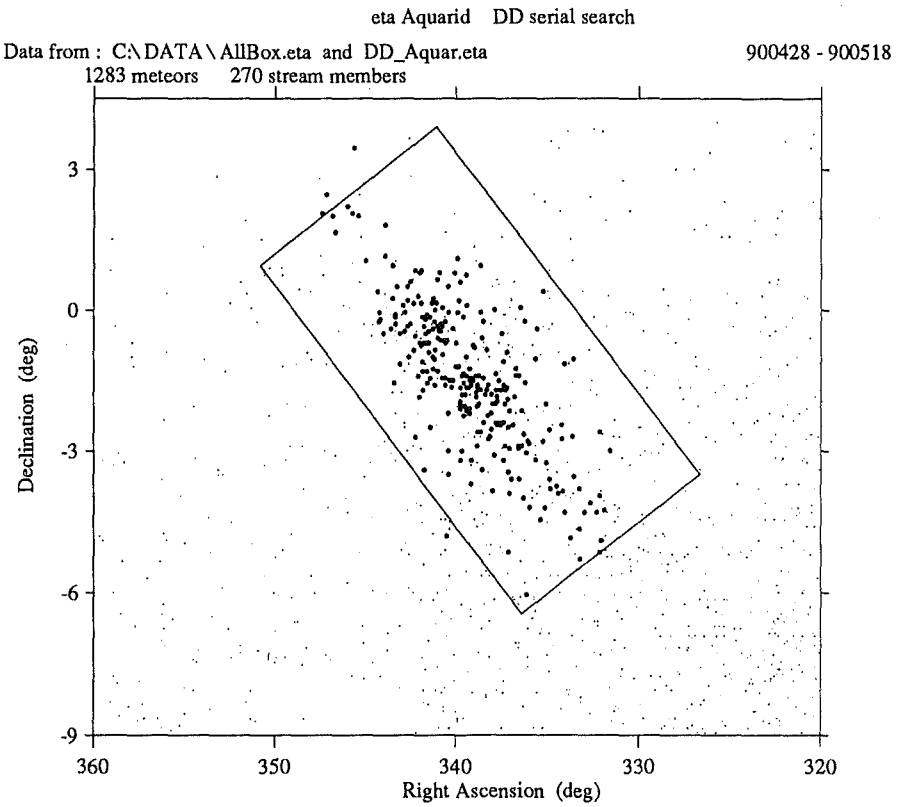


Figure 7.18: The radiants of stream members identified using a Drummond D-Criteria serial search are plotted as larger spots. The stream orbit from Section 7.4 is used to seed the group and subsequent membership required two parent sponsors within DD < 0.04 of the proposed orbit.

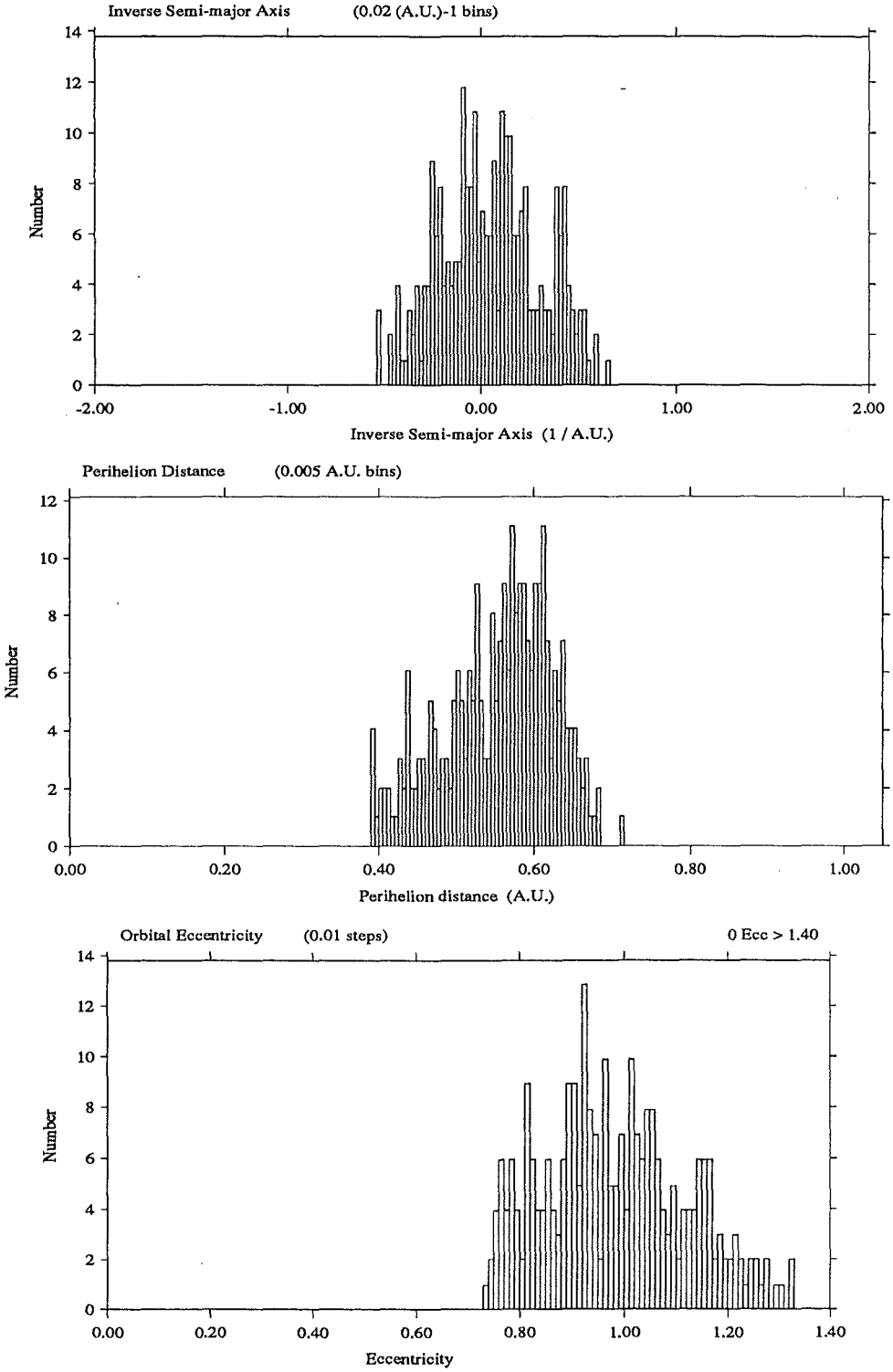


Figure 7.19: The orbital element distributions for 270 η Aquarid shower meteors selected using a serial D-Criteria search. Stream members must be within DD < 0.04 of at least two other members. The selection is seeded using the η Aquarid mean orbit from Section 7.2.

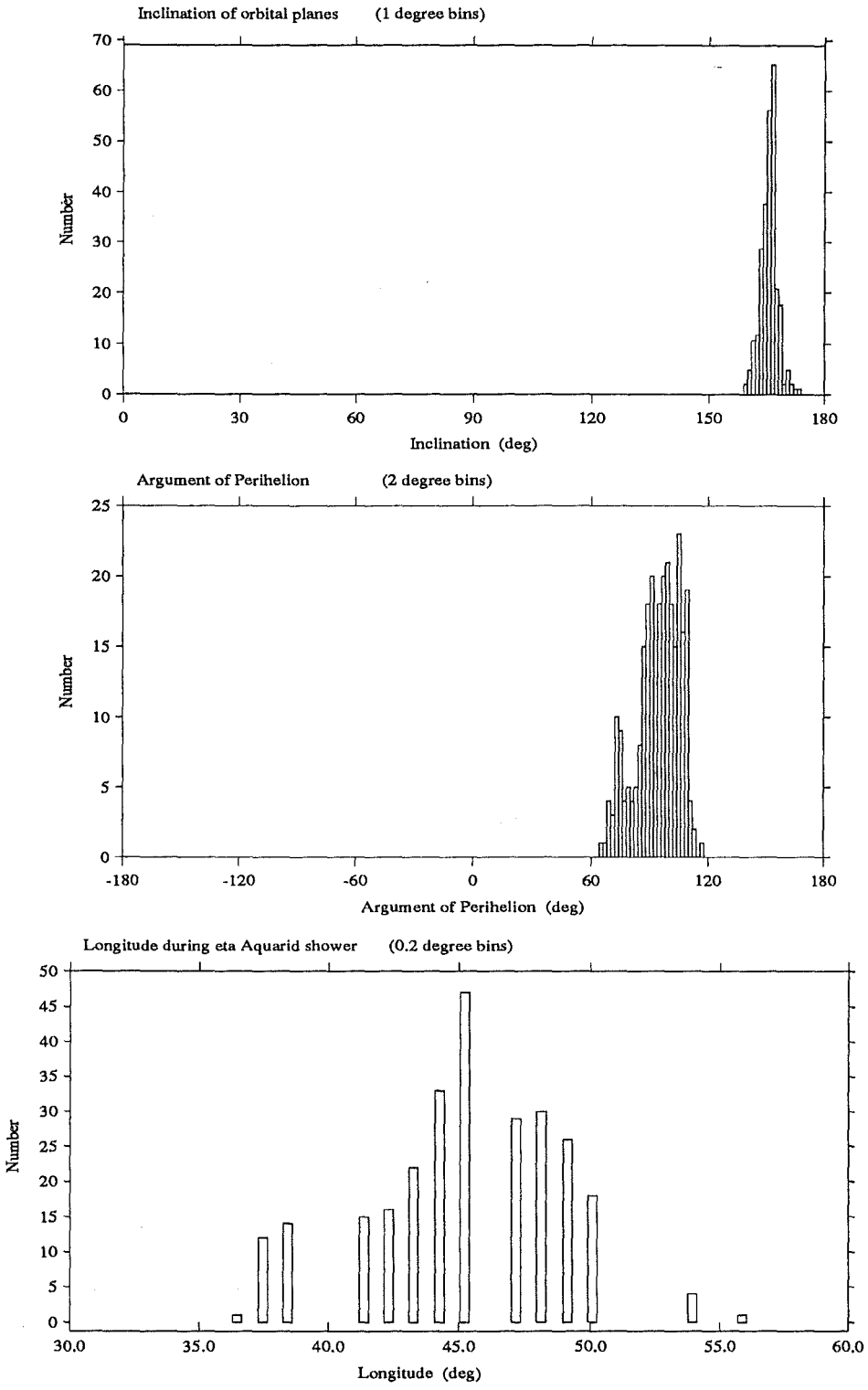


Figure 7.20: The orbital element distributions for 270 η Aquarid shower meteors selected using a serial D-Criteria search. Stream members must be within DD < 0.04 of at least two other members. The selection is seeded using the η Aquarid mean orbit from Section 7.2.

D-Criteria stream members, DD < 0.04, 2 sponsors.

	q	e	i	w	Ω_{1950}	V_m	Number
	0.555	0.983	165.5	94.5	45.5	41.3	270
s.e.	0.004	0.008	0.1	0.7	0.2	0.2	
s.d.	0.070	0.138	2.2	11.2	3.7	2.8	

7.6 Meteoroid Orbital Density

The mean orbit of the η Aquarid stream is established in section 7.4 by inspecting the orbital element distributions for shower meteors. The values of these stream elements are biased by having carefully chosen meteors with appropriate radiants and velocities. To substantiate the claim that this procedure yields a stream orbit, a check needs to be made that the orbital elements in fact represent a local maximum in the meteor orbit distribution. The number of observed meteor orbits within a particular level of similarity (DD < 0.10 for example) to a test orbit gives a measure of the meteoroid density around that orbit.

Each successfully reduced meteor observation represents a point in the five parameter orbital phase space. The Drummond D-Criteria gives a measure of the *distance* between points in this space. A series of cross sections are taken through the data around the proposed stream to evaluate the relative density of meteors in that orbit. Four of the orbital elements are held constant while the fifth is varied across the range of interest. The number of meteors within a given level of the Drummond D-Criteria is counted for each of the test orbits. The normalised density profiles are plotted in Figure 7.21.

The density profiles in Figure 7.21 come from two different binnings of the data. The more stringent DD < 0.03 curve shows a more pronounced peaking in density around the stream orbit. The peak density at the stream elements corresponds to 18 orbit observations within the D-Criteria range. At the more moderate level of similarity 165 observations have a DD < 0.10.

Each of the density profiles reaches a local maximum at the value of the corresponding stream orbital element. The data used to produce these profiles was not restricted to the radiant box used in selecting the shower meteors. The stream orbit is therefore confirmed.

Considerable caution needs to be exercised in interpreting the shape of these density profiles[5]. For example, as the eccentricity is moved away from the shower value the test orbit will shrink or expand. Holding the other four elements constant means the test orbit will no longer intersect with the Earth. Meteor orbits cannot be determined unless the particle hits the Earth. The curves of perihelion distance, eccentricity and argument of perihelion manage to prove this fact in a rather convoluted manner! As noted above, the profiles do confirm the choice of orbital elements for the stream. To eliminate this

[5]The double peak of the DD < 0.03 curve is within the statistical error for the sample.

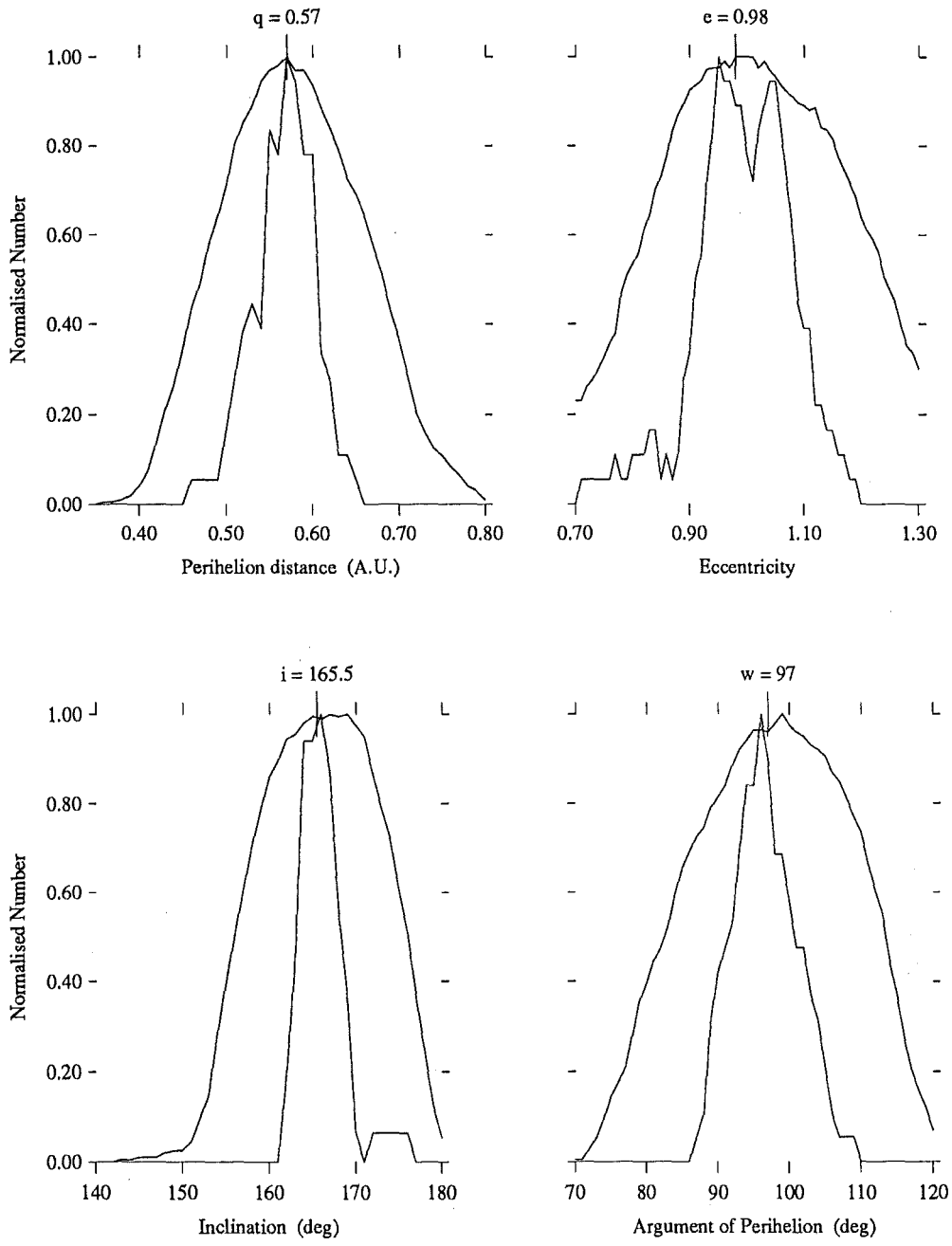


Figure 7.21: The population density of meteoroid orbits is scanned using two levels of the the Drummond D-Criteria (DD). Four of the elements are held constant on the η Aquarid orbit while the fifth is varied. The normalised number of meteor orbits within DD < 0.10 and DD < 0.03 is plotted for each cross section. The stream values chosen are also marked. The longitude of the ascending node is held constant for all four sections, $\Omega = 46°$.

problem density-profiles need to be taken where the test orbit intersects the Earth's orbit across the whole section.

Six parameters are necessary to define the position and velocity of a particle in space. This information can be transformed to five elements defining the orbit and a sixth, the time of perihelion passage which locates the object on that orbit. Meteor orbits are calculated from observations made during the destruction of the particle so the position of the object in its orbit is no longer relevant. The observation of individual meteor orbits is used as a probe to study the orbital distribution of meteoroid particles. In this sense, as well, the time of perihelion passage is also irrelevant. The six parameters defining the kinematic behaviour of the particle are reduced to the five necessary to define an orbit.

Since only Earth intersecting orbits produce meteors a restriction on those orbits which can possibly be detected is imposed. The observed data set has only four degrees of freedom. The position in space is fixed by the position of the Earth at the time of observation. The longitude of the Earth at observation identifies one degree of freedom. The three components of the heliocentric velocity the particle has before it approaches the Earth can be varied independently. This prescribes the other three degrees of freedom.

Appropriate cross sections can be taken by varying each of the four degrees of freedom. Test orbits corresponding to the chosen date of observation and heliocentric velocity vector are calculated. The number of observations within a defined level of the D-Criteria are counted as a measure of the density. This scheme ensures that any falloff in the density is not due to the test orbit moving away from a condition of intersecting with the orbit of the Earth. The Earth's motion, gravitational attraction, atmospheric effects and equipment parameters all conspire to ensure that all possible values of the heliocentric velocity are not equally likely. However making small variations about the characteristic velocity of the stream should not be heavily influenced by these factors. Variations in the longitude of observation are less affected by these factors. Varying the longitude forms the basis of the new stream search scheme suggested by Olsson-Steel (1987) and is discussed in the next section.

The heliocentric velocity is defined by three components. The specific components chosen for the density profiles across the η Aquarid shower are defined diagrammatically in Figure 7.22. These are the elongation of the true radiant from the sun, ϵ, the inclination of the orbital plane, i, and the magnitude of the heliocentric velocity, V_m. For the stream orbit identified in this chapter the η Aquarid heliocentric velocity when intersecting the Earth is given by; $i = 165.5°$, $\epsilon = 41.0$ and $V_m = 41.6\text{km.s}^{-1}$. The velocity can be expressed in cartesian coordinates.

$$\mathbf{v}_m = -V_m \begin{pmatrix} \sin\epsilon\cos i \\ \cos\epsilon \\ \sin\epsilon\sin i \end{pmatrix} \tag{7.12}$$

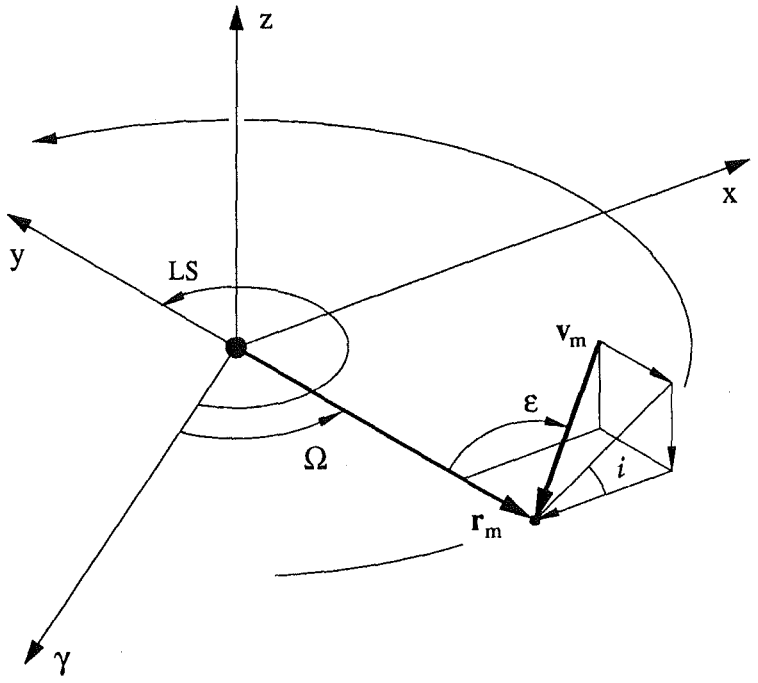


Figure 7.22: Meteor orbits that intersect the Earth have four degrees of freedom. These can be expressed as the longitude of the ascending node, Ω, and the three components of the heliocentric velocity. For the D-Criteria density profiles the three velocity components chosen are: Solar elongation of the true radiant, ϵ, inclination of the orbital plane, i and heliocentric speed.

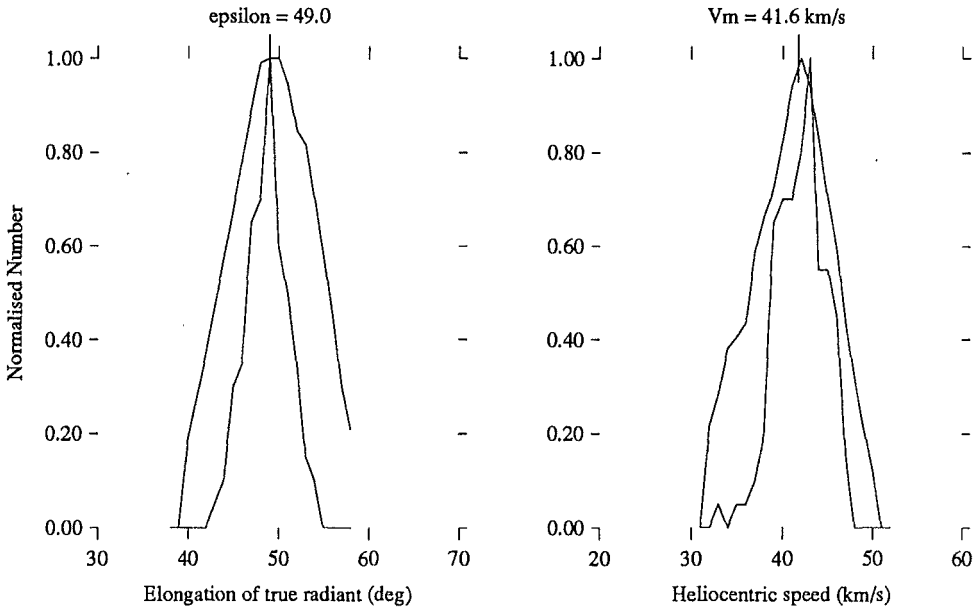


Figure 7.23: The population density of meteoroid orbits is scanned using two levels of the the Drummond D-Criteria (DD). All the test orbits intersect the Earth. One component of the heliocentric velocity is varied whilst the other two are held constant. The corresponding orbit for such a meteor observed at the descending node ($\Omega = 46°$) is used. The normalised number of meteor orbits within DD < 0.10 and DD < 0.03 is plotted for each cross section. The maximum numbers included are 164 and 18 respectively. The velocity components corresponding to the stream orbit are also marked.

The density profile obtained by varying the inclination was calculated while checking the proposed orbital elements for the η Aquarid stream. Given any meteor orbit then varying the inclination will still produce an orbit which intersects the Earth. The inclination has been chosen as one of the heliocentric velocity components for this reason. The enhancement of meteors with inclinations around 165.5° is the most pronounced feature of the shower. If the meteoric material was deposited during the few perihelion passages leading up to the 1066 AD apparition of comet Halley then the stream might be expected to lie in the same plane as the comet at that epoch. Yeomans and Kiang (1981) calculate comet Halley had an orbital inclination of 163.4° at that time. Gravitational perturbations due to planets near the plane of the ecliptic might be expected to draw dust particles toward less inclined orbits. The mean orbital plane of the stream seems to have moved two degrees toward the ecliptic in the thousand years since the comet was last in a corresponding orbit that intersected the Earth.

The density profiles associated with the other two components are plotted on Figure 7.23. The cross section in solar elongation shows a definite enhancement at the expected stream value. The profile taken by varying the speed of the orbit at detection peaks just under the solar escape velocity. Both these profiles confirm that there is a definite increase in the number of meteor orbits around the η Aquarid stream orbit.

Two curves have been used to sketch the D-Criteria orbit cross sections. The less strict criterion of DD < 0.10 includes orbits of meteors less tightly associated with the stream. The contour for this level will sample more meteors that have been slightly perturbed away from the stream orbit. The speed profile in Figure 7.23 shows more meteors in this dispersed group at lower velocities. Radiative perturbation effects would cause a meteoroid to lose kinetic energy and this curve provides tentative evidence of this. For a discussion of the Poynting-Robertson effect see Burns *et al* (1979). The broad variation in heliocentric velocity profile may be due to uncertainties involved in measuring the meteor velocity and then correcting this to that of the meteoroid before entry into the atmosphere. Measurement uncertainties from Section 7.4 indicate a range of values from 35.8 to 48.6 km.s^{-1}.

The D-Criteria calculations were done using the data subset plotted in Figure 7.1. Some of the tailing off in density of meteors as the test orbit is moved away from the stream may be due to this artificial imposed cutoff. A test was conducted using all the data available with the heliocentric speed profile. The difference between the limited radiant group and the full set was less than half the implied statistical error for two extreme points in the cross section. Otherwise the results were identical and took about ten times as long to compute.

The six density cross sections discussed in this section all use a value for the longitude of the ascending node that corresponds to the middle of the shower activity. Any simple cross section taken through the longitude will be heavily biased by the particular days in which the shower was observed and the length of time spent successfully observing on those days. Eliminating these biases is discussed in the next section. Conducting a scan across the longitude of the ascending node is the basis of the D-Criteria stream search proposed by Olsson-Steel (1987).

7.7 D-Criteria Stream Search

A concentration of meteor orbits around the η Aquarid stream orbit has been found in the AMOR data. This concentration should be compared with observations made when the shower was not present. To confirm the existence of the shower it should be shown that at other times of the year significantly fewer meteors come from the same relative part of the sky at the same relative speed. Equipment and data reduction effects for the system will be similar for

meteors coming from the same relative direction making the comparison more valid. The speed and radiant fix the perihelion distance, eccentricity, inclination and argument of perihelion whilst the date locates the longitude of the ascending node[6]. If the data set were entirely made up of random orbits then the number of correlated orbits for any value of the ascending node would be roughly similar. Where a stream is present a significant enhancement should be evident at the appropriate longitude.

This stream search technique was proposed by Olsson-Steel (1987). It was introduced to find meteor streams associated with Earth crossing asteroids in the Adelaide meteor orbit data. This data set contained 3759 orbits collected using an all-sky radar system and included reasonably even coverage during the year. The search was conducted by assigning a nodal longitude to the test orbit and counting the number of orbits within a D-Criteria similarity of 0.125. The data set collected using the AMOR system did not have the uniformity of observing coverage necessary for an application of the stream search in this form. The daily data rate of around 1500 observations does allow considerable flexibility in restricting the search to eliminate the bias from *patchy* observing.

During the period of the 1990 η Aquarid shower the radar was run daily. Prior to that two twenty-four hour observing runs had been obtained at near monthly intervals. To ensure that the stream-search did not merely reflect the intensive observing program during the shower only those observations made on individual days were included in each count. Back-scatter sea echoes reflected off sporadic-E ionisation usually stopped effective observing at some time between 0700 and 1000 NZST. The η Aquarid stream transits at around 0730 hours, so for days where the sporadic-E was less obstructive many more meteors could be detected from that region of the sky. Uniformity was maintained by only including those observations made between 0300 and 0700 hours. For any observation in this sample the contribution of the longitude of the ascending node to the value of the D-Criteria was reduced to zero by setting the test nodal longitude equal to the solar longitude of the observation. Since the η Aquarid shower results from particles moving through their descending node this is equivalent to setting the test node equal to the ascending node of the observation. It does however exclude Orionid meteors, particles from the same stream but detected at the ascending node. The longitude was effectively restricted by counting only those with the appropriate similarity in the four hours prescribed on any individual day observations were made.

Figure 7.24 shows the results of varying the longitude for three different values of the Drummond D-Criteria. The orbital elements used were those chosen for the stream. The profile shows a significant enhancement at the longitude of the shower for all three levels of similarity. This provides another confirmation that the concentration of meteors belongs to a meteoroid stream of astronomical origin.

[6]This assumes that the Earth's orbit is circular

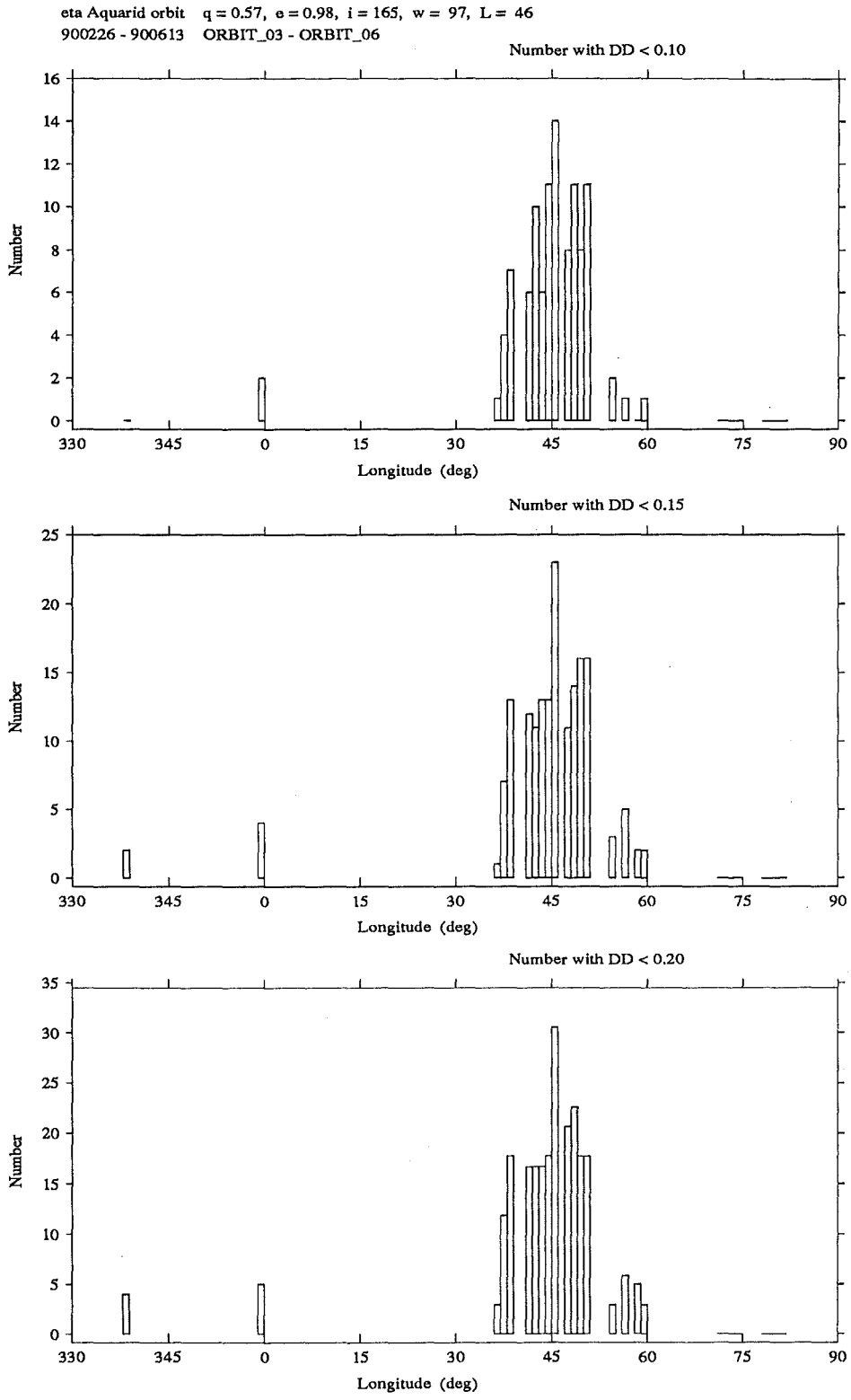


Figure 7.24: The number of meteor orbits correlated with the η Aquarid stream. Only those meteors detected between 0300hrs and 0700hrs are included. The longitude of the ascending node is adjusted to be the same as the solar longitude at the time of observation. The number within the prescribed D-Criteria similarity for each day is plotted.

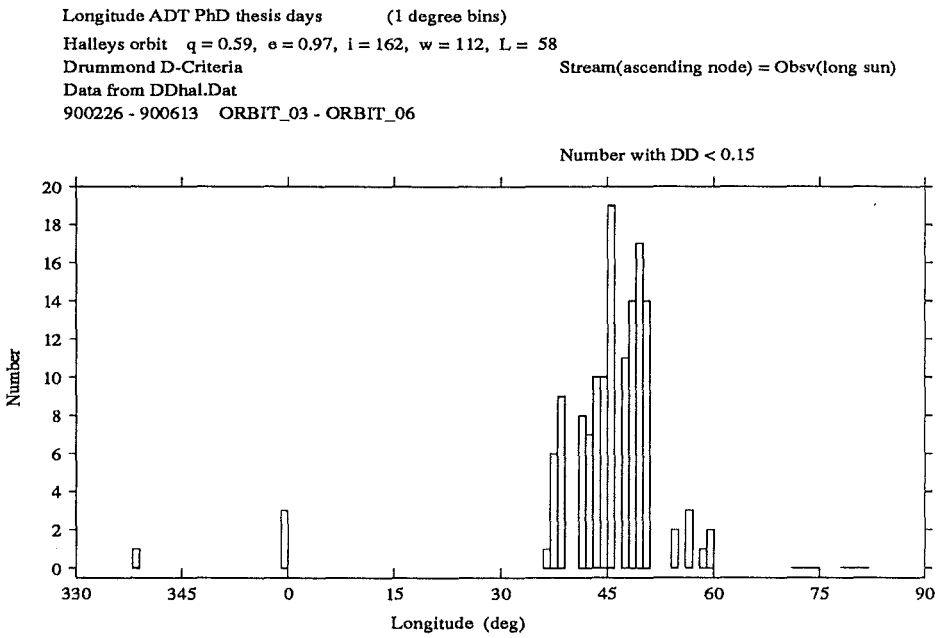


Figure 7.25: The number of meteor orbits correlated with the orbit of comet Halley. Only those meteors detected between 0300 and 0700 NZST are included. The longitude of the ascending node is adjusted to be the same as the solar longitude at the time of observation. The number of meteor orbits with DD < 0.15 on each day is plotted as a function of longitude.

Following the procedure of Olsson-Steel the orbital data collected between February and June 1990 has been used to search for a stream associated with Comet Halley. The results are plotted in Figure 7.25. As expected, the AMOR data shows a significant enhancement near the longitude of the comets ascending node supporting the claim that the meteoroid stream is associated with Halley.

7.8 Fine Structure ?

In all the radiant plots presented in this chapter it is possible to see some additional fine structure *clumping* in the stream. The small boxes overlaid on Figure 7.26 provide boundaries on the centre of these clumps. The mean orbital elements and standard deviations for the stream members that fall within each of these boxes have been calculated. Each of the clumps contain approximately the same number of meteor orbits.

North Clump, $\alpha = 341.5°$, $\delta = -0.4°$.

	q	e	i	w	Ω_{1950}	V_m	Number
	0.535	0.972	165.6	91.7	44.6	40.9	65
s.e.	0.009	0.017	0.1	1.5	0.2	0.4	
s.d.	0.073	0.141	1.2	12.2	1.6	3.0	

	q	e	i	w	Ω_{1950}	V_m	Number
South Clump,	$\alpha = 339.0°$,		$\delta = -1.8°$.				
	0.575	1.013	165.5	97.8	48.1	42.0	61
s.e.	0.007	0.015	0.1	1.1	0.2	0.3	
s.d.	0.053	0.115	1.1	8.7	1.4	2.2	

The two groups contain meteors with two slightly but distinctly different sets of orbits. The northern clump with a slightly lower average velocity shows a larger dispersion of orbital elements amongst its members. McIntosh and Hajduk (1983) introduced a shell model to explain the η Aquarid and Orionid meteor showers. The progenitor comet will eject meteoric particles at each perihelion passage which will rapidly spread around the orbit. They argue that as the parent comet's orbit evolves this will lay down a shell of particles. The shell will then thicken to form a belt. The meteor showers occur where the Earth moves through the belt. The orbit of the comet continually changes due to planetary perturbation and mass ejection. Repetitions of this libration cycle would produce a set of overlapping belts possibly separated by slight perturbation in the comet's motion. The two clumps discerned in the 1990 AMOR observations fit such a model. The suggestion here is that the slower North clump with its more dispersed elements would be the older more evolved belt. The tighter South clump would be associated with material deposited in the most recent libration cycle.

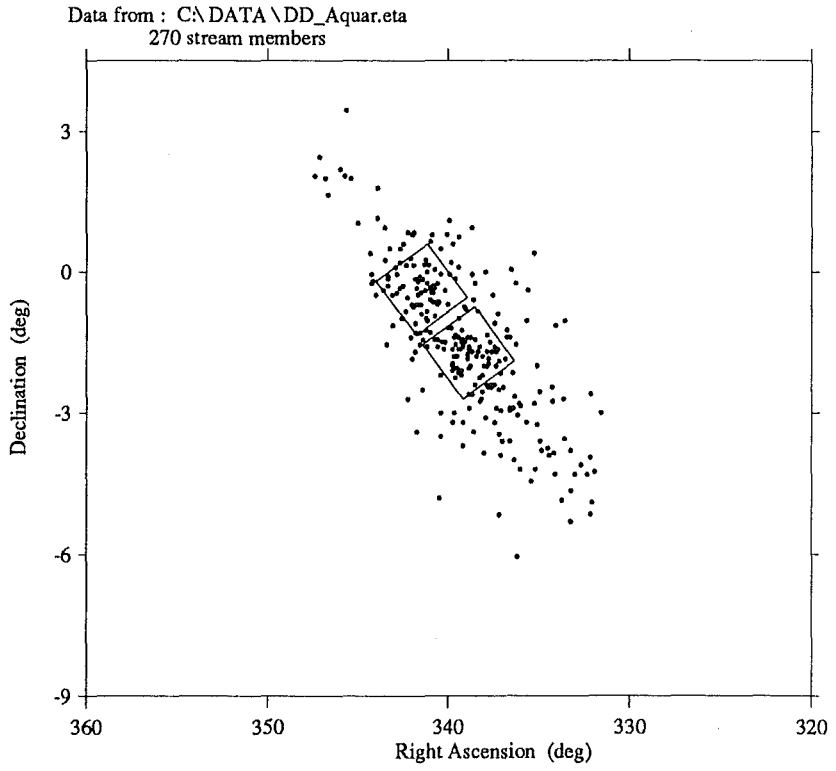


Figure 7.26: Two clumps are evident in the radiant distribution of the η Aquarid shower meteors. The D-Criteria serial search stream members are plotted with small radiant boxes to select members from the centre of the clumps.

McIntosh and Hajduk (1983) interpret the meteor rate data for the η Aquarid shower as a set of belts each with a duration of peak activity of about five days. For each of the radiant clumps identified above the standard deviation in the longitude of the ascending node is 1.5°. If the belt were uniformly distributed across the nodal longitude this would correspond to a $4\frac{1}{2}$ day width. The two clumps seem to be individual belts within the shell model.

It should be noted that no observing run was carried out on May 8 (Day number 127). The batteries supplying power to the remote receiver sites required more complete recharging after six days of operation. The missed day ($L_{\odot\ 1950} = 47°$) falls right in the middle of the longitude distribution of the two clumps. It may be that this would fill in some of the distinctive gap between them. The clumping into two radiant sources can be seen in daily shift of the mean radiant, Figure 7.7. That day 128 does not lie between 126 and 129 suggests the May 8 mean radiant could be anywhere in the region. The two mean radiants from April 29 and 30 (118, 119) could be part of a third belt. Again observations from May 1 and 2 would clarify the situation.

7.9 Summary of the η Aquarid Stream

The η Aquarid meteor shower has been used to calibrate the Christchurch AMOR system. This section summarises the 1990 results. Table 7.4 tabulates the orbital elements for the stream.

Shower members were identified using two methods. A radiant box aligned parallel to the ecliptic together with a heliocentric velocity selection provided 361 members, Section 7.2. A serial search for members using the Drummond D-Criteria giving 270 members was also conducted as a comparison. The distribution of orbital elements for both methods were similar. Mean values for the two sets of stream members are found in Table 7.4. Both are in good agreement within the standard error of the mean values. As is evident from the diagrams showing the orbital distributions, the D-Criteria members show less dispersion. They represent a sample of orbits more closely associated with the mean. This may reflect less contamination from sporadic meteors. The mean orbital information derived seems to be independent of the membership selection criteria chosen.

Using mean orbital elements to describe the stream is sensible where there is a reasonably symmetric distribution. The substantial data set available allows the actual distributions to be inspected and peak values chosen to describe the stream orbit. These values agree reasonably with the calculated means. They are included in Table 7.4 as final summary values for the 1990 observations. Using the D-Criteria a series of orbital density cross sections were taken. These confirm the stream orbit.

The distribution of orbital elements for the shower closely follow that expected from measurement uncertainties in the station data. The measured

η Aquarid orbital elements

	q	e	i	w	Ω_{1950}	V_m	Number
Stream Orbit, AMOR Christchurch 1990.							
	0.57	0.98	165.5	97	46	41.5	270
Comet Halley, Yeomans and Kiang (1981).							
1986 AD	0.587	0.967	162.2	111.8	58.1		
989 AD	0.582	0.968	163.4	101.5	45.8		
607 AD	0.581	0.968	163.5	98.7	42.5		
A working list of meteor streams, Cook (1972).							
	0.560	0.958	163.5	95.2	42.4	41.3	1
Summary of orbits, Lindblad (1989).							
Photographic	0.612	0.983	165.5	101.5	45.8	41.7	11
Radar	0.584	0.882	165.7	95.9	45.5	39.9	5
Radiant-Velocity selection, Radiant box, $34 < V_m < 48$ km.s^{-1}							
	0.566	0.969	165.8	95.4	45.8	41.0	361
s.e.	0.006	0.009	0.1	0.9	0.2	0.2	
s.d.	0.118	0.171	2.8	16.7	4.1	3.3	
D-Criteria stream members, DD < 0.04, 2 sponsors.							
	0.555	0.983	165.5	94.5	45.5	41.3	270
s.e.	0.004	0.008	0.1	0.7	0.2	0.2	
s.d.	0.070	0.138	2.2	11.2	3.7	2.8	
North Clump, $\alpha = 341.5°$, $\delta = -0.4°$.							
	0.535	0.972	165.6	91.7	44.6	40.9	65
s.e.	0.009	0.017	0.1	1.5	0.2	0.4	
s.d.	0.073	0.141	1.2	12.2	1.6	3.0	
South Clump, $\alpha = 339.0°$, $\delta = -1.8°$.							
	0.575	1.013	165.5	97.8	48.1	42.0	61
s.e.	0.007	0.015	0.1	1.1	0.2	0.3	
s.d.	0.053	0.115	1.1	8.7	1.4	2.2	

Table 7.4: Orbital elements for the η Aquarid stream. These are mean elements calculated from those meteor orbits identified as belonging to the stream (or component). The stream orbit adopted for the AMOR 1990 data instead uses the peaks in the distributions of the orbital elements. The number of meteor orbits included in the determination of each orbit is noted. The standard error in the mean (s.e.) is the standard deviation (s.d.) divided by the square root of the number of orbits.

radar system parameters for a typical shower meteor were varied randomly within a normal distribution with one standard deviation estimates of the errors. The orbital elements for a sample of 6000 trails reflected both the shape and limits of the stream distributions. The η Aquarid stream members could all come from the same mean orbit within the estimated limits of measurement uncertainties.

The enhancement of observations around the stream orbit was investigated to ensure that it was not merely a product of selection, reduction and observational biases. Orbital density profiles were taken where each test orbit in the cross section intersected the Earth. Varying the three components of the heliocentric velocity (inclination, elongation of the true radiant and speed) all showed the density of observed orbits peaking around the expected value for the stream. The D-Criteria search technique based on a cross section through the ascending node also revealed a stream at the longitude of the η Aquarid shower. A similar cross section using the Halley orbit again confirmed an association with the stream's parent comet.

The background distribution of sporadics left once η Aquarid meteors are removed shows a smooth variation in radiants. This is consistent with that expected from the configuration of the radar and the concentration of particles in the ecliptic. The distribution of orbital elements for the sporadic complex in the region of the shower radiant showed an excess of particles with the stream inclination. This group could represent particles that have been strongly perturbed from the stream orbit but retained their orbital inclination. The inclination for η Aquarid observations is the element least affected by measurement uncertainties. Timing uncertainties render the other elements error-prone while only affecting the inclination slightly. An anomalous number of persistent meteors observations grouped around the shower radiant remained amongst the non-shower data. Precise timing of the rising edge of long persistent echoes is often ambiguous and may contribute to increased timing uncertainties for these meteors.

The orbital elements in Table 7.4 can easily be compared with that of previously published results. They agree astoundingly well. The number of orbits for shower meteors has been substantially increased.

As the Earth moves through the η Aquarid shower the radiant also moves across the sky. The mean radiant for shower meteors observed in each day was calculated. These showed a daily shift parallel to the ecliptic. This was in agreement with Lindblad's radiant ephemeris, Cook's estimate of daily motion and the observations in New Zealand of McIntosh (1934). This represents the first demonstration of daily motion in the radiant using radar meteor observations.

The radiant distribution shows two distinct clumps. Inspecting the orbital elements for the two groups shows them to be significantly different. Their duration of around five days tentatively identify them as individual belts in the shell model proposed by McIntosh and Hajduk. The slightly higher velocity

and smaller dispersion in the orbital elements of the South Clump suggest this is the more recently formed less dispersed belt.

A stream orbit was obtained from the 1990 AMOR observations of the η Aquarid shower. The orbit obtained agreed well with previous measurements by other meteor orbit systems. The overall performance of the AMOR system is validated. The increase in data on the stream is substantial. Data is now available for the direct study of stream evolution. The data set includes evidence supporting the McIntosh and Hajduk shell model of the Halley meteor stream.

Chapter 8

Overview of AMOR

8.1 AMOR Data Distributions

8.1.1 Perihelion Distances

The distribution of perihelion distances in Figure 8.1 shows a peak in the number of orbits detected near perihelion at 1 AU. Particles near their perihelion or aphelion spend relatively longer at a given radius from the Sun than at any other time during their orbit. Where this is the same as the orbital radius of the Earth the likelihood of a collision is increased. This increase is well understood and its influence summarised by the cosmic weighting factor introduced by Whipple (1951).[1]

In fact the enhancement of orbits with a perihelion distance near 1 AU in Figure 8.1 shows a double peak. The orbit reduction software used by AMOR allows for the orbital eccentricity of the Earth, see section 6.8. The heliocentric position of the meteoroid is assumed to be that of the Earth and so the position of this cosmic weighting peak will reflect the relative time spent by the Earth at that heliocentric distance from the Sun. Table 8.1 summarises the number of days the Earth spends in each of the orbital radius bins. Because of the actual orbital dynamics of the Earth more time, in total, is spent within 0.990 AU and outside 1.010 AU making a double peak to the distribution of perihelion distances likely.

In fact the outer-most peak of Figure 8.1 is between 1.000 and 1.010 AU. The situation is complicated by the actual time spent observing at each of the different distances. A summary of the number of days with reasonable coverage during the morning, when most meteors are detected, is included in Table 8.1. Relatively fewer observations were obtained between June and November before the 40 Amp.Hour batteries were purchased to power the

[1] Referring to Figure 8.7 it can be seen that within the AMOR data set the enhancement of orbits with a perihelion distance near 1 AU should be attributed to those meteoroids detected in retrograde orbits. The same cosmic weighting mechanism is evident in the increase in number of prograde meteors detected near aphelion.

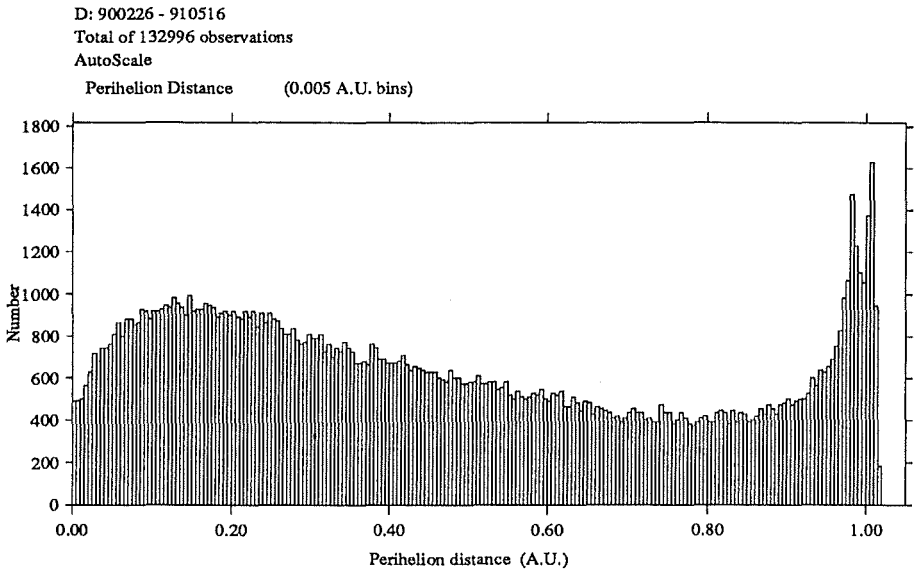


Figure 8.1: The perihelion distances for meteors orbits observed by AMOR between 1990 Febuary 26 and 1991 May 16. The enhancement in observed orbits with a perihelion near 1 AU is double peaked. The increase in the number of orbits with perihelion distances nearer to the Sun is probably the result of a selection effect against prograde meteoroids observed near perihelion.

Details of days in radius bins

Dates of Year			Radius (AU)	Annual	1990	1991
January 1	-	February 27	0.983 - 0.990	58	1	18
February 28	-	April 5	0.990 - 1.000	37	1	12
April 6	-	May 12	1.000 - 1.010	37	12	26
May 13	-	August 29	> 1.010	109	23	4
August 30	-	October 5	1.010 - 1.000	37	10	
October 6	-	November 11	1.000 - 0.990	37	6	
November 12	-	December 31	0.990 - 0.983	50	15	

Summary of days in radius bins

Radius	Annual	AMOR
< 0.990	108	34
0.990 - 1.000	74	19
1.000 - 1.010	74	48
> 1.010	109	27

Table 8.1: The number of days the Earth spends in the series of heliocentric radius bins associated with the perihelion distance plot of Figure 8.1 A double peak related to the cosmic weighting is expected when the orbital eccentricity of the Earth is included in the meteor orbit reduction scheme. The number of days spend observing with AMOR that gave reasonable morning coverage in the same periods is also included.

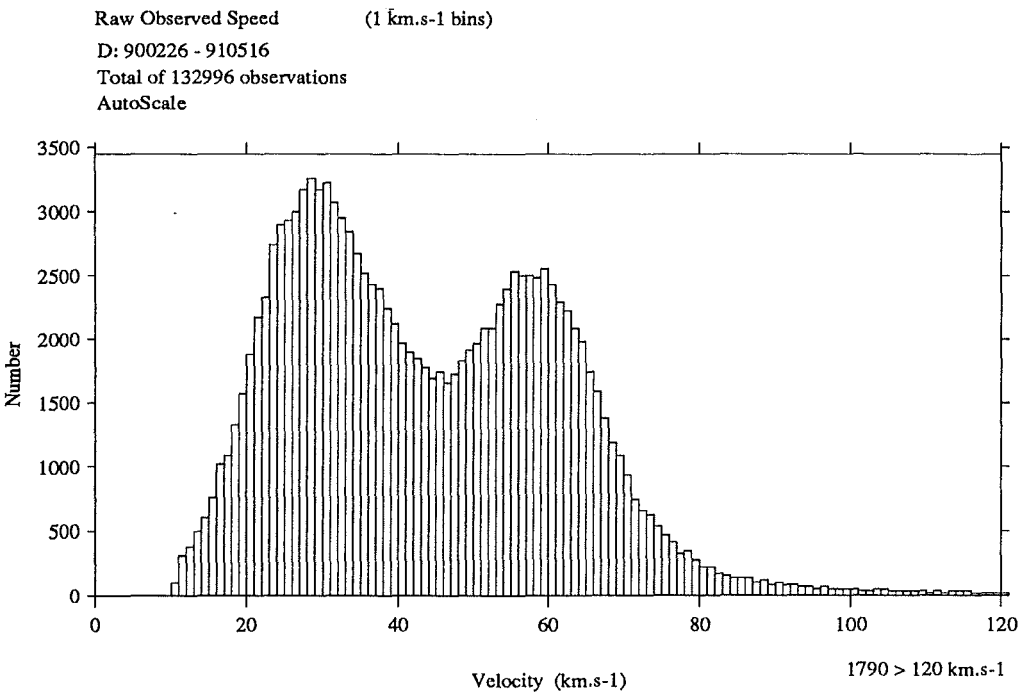


Figure 8.2: The distribution of measured atmospheric meteor speed shows a bimodal distribution.

remote receiver sites and data links. At dawn from September to November the ecliptic moves well north of the AMOR radar beam thus reducing the number of meteors detected. Observations during the period of the η Aquarid meteor shower in early May from both 1990 and 1991 are included in the data set. This month-long period is the only part of the orbit that has been observed extensively more than once.

8.1.2 BiModal Velocity Distribution

The distribution of observed meteor velocities in Figure 8.2 shows a distinctly bimodal distribution. This phenomenon is a result of the 30 km/s orbital motion of the earth and the heliocentric orbits of the meteoroids. Meteors travelling in retrograde orbits will tend to hit the Earth head-on adding their velocity to that of the Earth producing a relatively large observed atmospheric velocity. The low velocity of prograde meteoroids relative to the Earth produces the low velocity peak. Particles in 'orbits' intermediate between these two groups are falling directly toward the Sun making a completion of more than one such orbit very improbable.

From the AMOR data base it is relatively easy to produce a graphical display showing this explanation of the bimodal velocity distribution. The

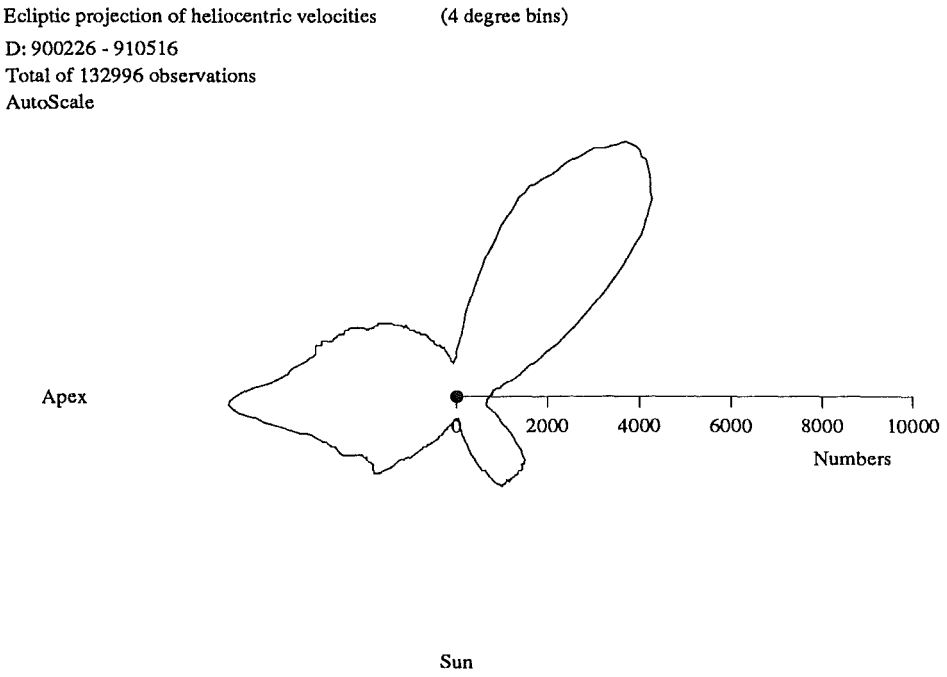


Figure 8.3: Ecliptic projections of the heliocentric velocities. The number of meteoroid trajectories coming from that direction in space are plotted. Retrograde meteors come from the direction of the Apex of the Earth's Way while prograde meteoroids are shown catching up with the Earth. Midday and afternoon observations were frequently precluded during the period of observations plotted with the result that many more prograde meteors are detected moving toward the Sun rather than away.

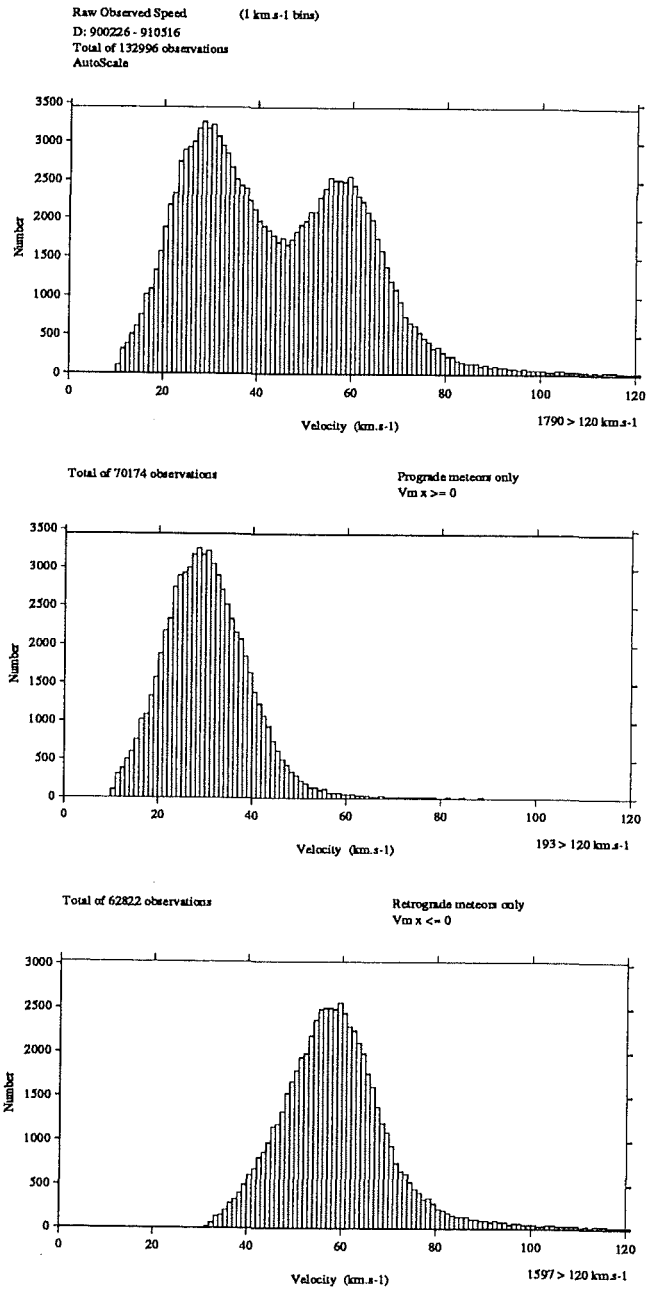


Figure 8.4: The components of the bimodial distribution of observed meteor speeds (top) are identified with the two distinct groups of prograde (middle) and retrograde (bottom) meteoroids detected.

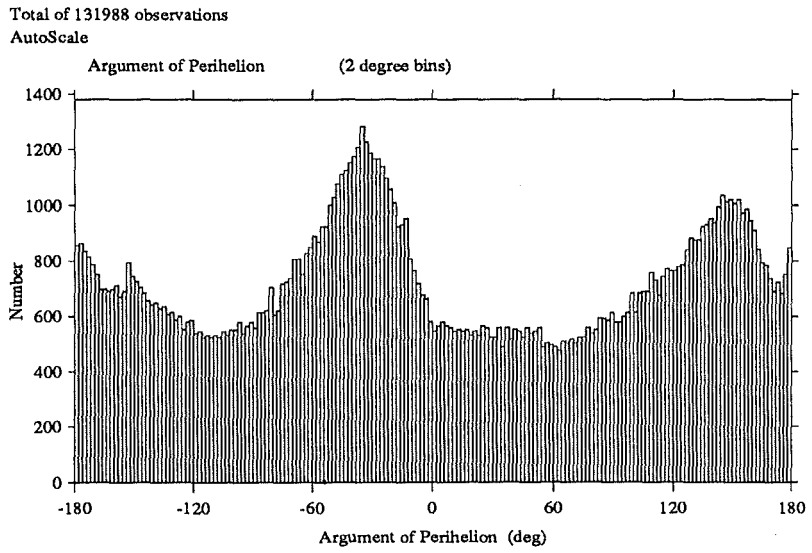


Figure 8.5: A cumulative distribution for the argument of perihelion. The plot shows an enhancement of meteors with argument of perihleion at $\omega \sim -35°$ and 150°.

ecliptic projection of the heliocentric velocities, Figure 8.3, shows the clear separation into prograde and retrograde groups with none going directly toward or away from the Sun. Figure 8.4 shows the graphical output from three runs through the data. The first reproduces Figure 8.2 showing the observed meteor velocities for all observations. The second panel plots the observed speed distribution of all prograde meteors detected and the final run shows the retrograde component producing the high velocity mode.

8.1.3 Asymmetric Argument of Perihleion

The cumulative plot for the argument of perihelion shows a broad peak in the distribution centred at -35° and 150°, see Figure 8.5. The argument of perihelion measures the angle centred on the Sun around meteor orbit from the longitude of the ascending node to perihelion. For a particle detected at its ascending node a negative argument indicates it was detected while moving away from the Sun, after perihelion passage. For a given orbit, detection before or after perihelion passage would be equally likely, producing a distribution for the argument of perihelion symmetric about 0° and 180°. The asymmetry must be the result of a selection effect within the radar system.

Figure 8.6 plots the argument of perihelion for the data set split into ret-

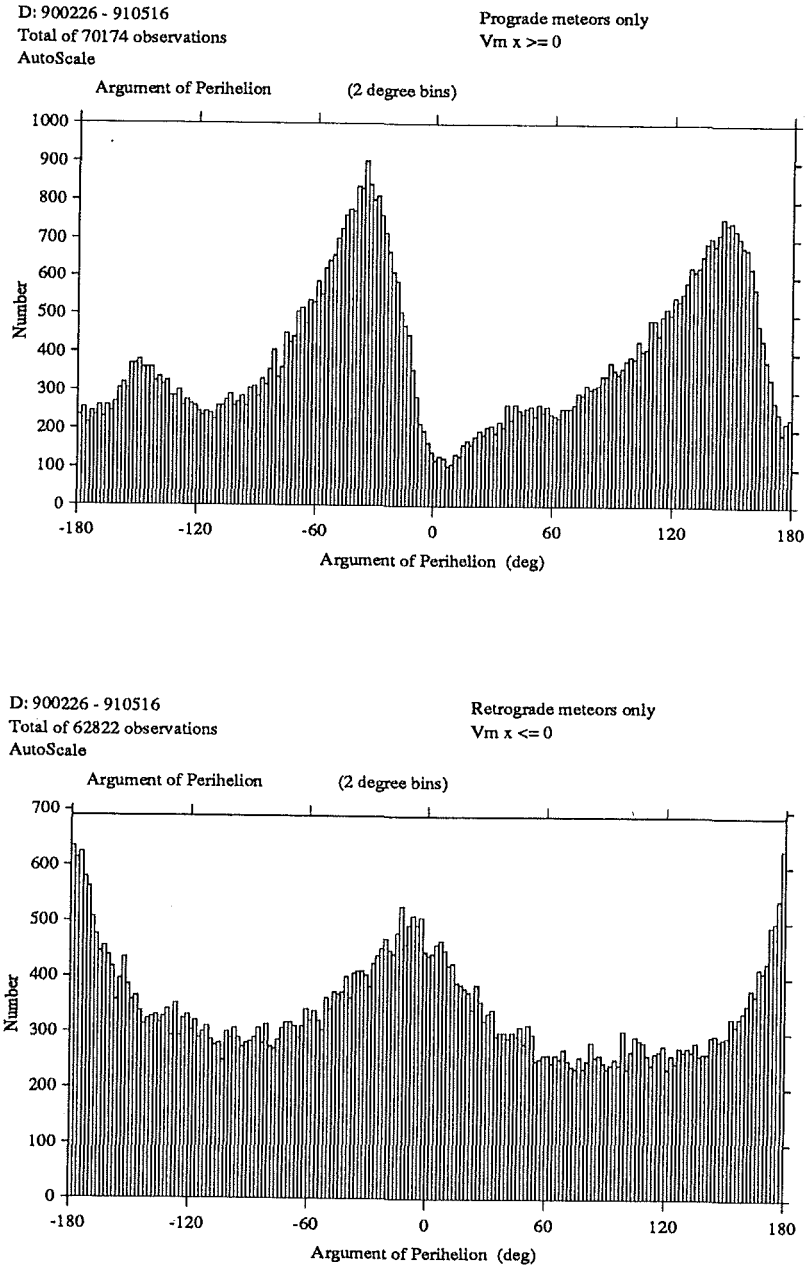


Figure 8.6: The argument of perihelion is plotted for the prograde and retrograde groups of meteor orbits separately. The top distribution shows the marked asymmetry in the argument of perihelion for meteors detected in prograde orbits.

rograde and prograde components. As expected, the distribution for meteors in retrograde orbits is symmetric. Figure 8.3 shows roughly equal numbers of meteors in retrograde orbits detected moving in toward and away from the Sun. Sea and land surface echoes reflected off sporadic-E ionisation during the day produces substantial interference, in many instances making meteor observations with AMOR impossible. Very few orbits have been obtained between 0900 and 1800 hours when prograde meteors moving away from the Sun would be detected compared with the period between 1800 and 0300 hours. Figure 8.3 confirms this situation where the number of prograde meteors moving toward the Sun are greatly enhanced compared with those moving from perihelion.

Understanding the distribution is considerably complicated by the fact that meteoroids have their argument of perihelion shifted by 180° depending on whether they are detected at their ascending or descending nodes.

Figure 8.7 shows the distribution of perihelion distances for meteors in prograde orbits. Because of the 30 km/s prograde velocity of the Earth, very few prograde meteors near perihelion have a sufficiently large relative velocity to produce an ionisation trail that can be detected by the radar. The vector geometry means that prograde meteoroids near aphelion can still be detected. Six times more prograde meteors are detected within 0.3 AU of aphelion than within the same distance of perihelion[2]. The increased collision probability of the cosmic weighting and a greater density of dust orbits with smaller semi-major axis may both be contributing factors in this enhancement of meteors detected nearer aphelion than perihelion.

Many more prograde meteors are detected nearer to aphelion than perihelion. These are usually detected shortly after aphelion since daytime observations of meteors moving away from the Sun were precluded during most of 1990 and 1991 by sporadic-E. Prograde meteoroids detected at their ascending node just after aphelion will have an argument of perihelion, $\omega \sim 150°$. Those detected at their descending node just after aphelion have an $\omega \sim -30°$. The asymmetery in the distribution of the argument of perihelion is a product of the lack of daytime observations combined with the dearth of prograde meteors observed near perihelion from the orbiting Earth.

8.2 Upgrades to AMOR

The Christchurch AMOR radar has been in successful routine operation for over a year and has determined more than 1.3×10^5 high grade meteor orbits. AMOR works and produces reliable orbits. The attendant advantages and disadvantages of possible upgrades to the system are discussed in this

[2]The program `DistPlay.Pas` was used to count 1251 prograde ($V_m x < 0$) near aphelion ($q' - R_m < 0.3$) meteors compared with 200 near perihelion ($R_m - q > 0.3$) for the interval 1990 Feb 26 to 1990 June 4.

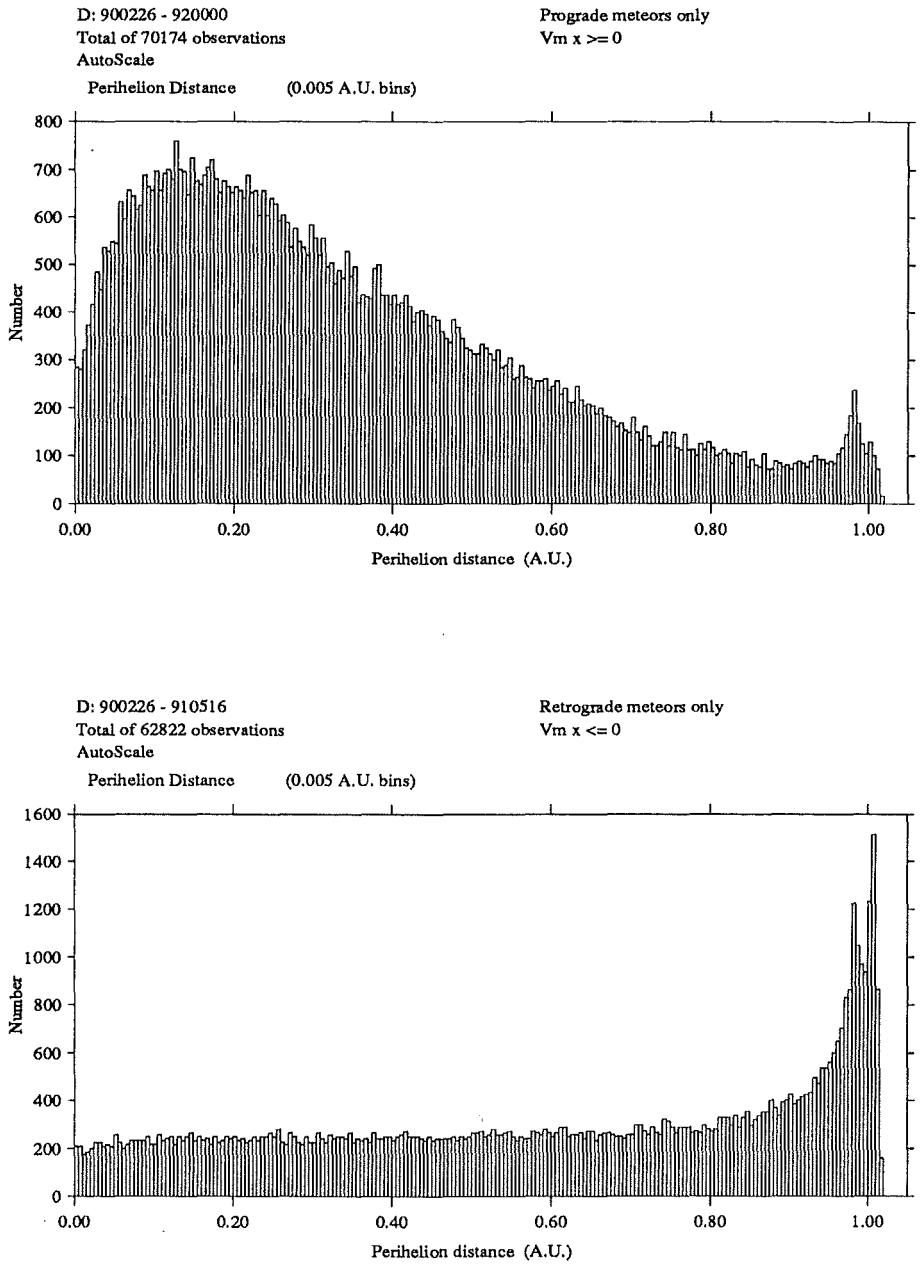


Figure 8.7: The perihelion distances for meteors detected in prograde and retrograde orbits. Most prograde meteors are detected well away from their perihelion. For retrograde orbits there is quite different more uniform distribution with an enhanced collisional probablity for perihelion distances similar to the orbital radius of the Earth.

section. Some more fundamental limitations of an orbit radar of this type are emphasised.

8.2.1 Narrower Beam

By building longer aerials it should be possible to narrow the effective radar beam used by AMOR. A narrower beam would provide greater accuracy in implicitly locating the meteor trail. The increased angular precision in conjunction with the specular reflection condition would fix the plane in which the meteor was moving with greater certainty. A narrower beam would also increase the antenna gain thereby detecting fainter meteors.

Meteor echoes are required at all three receiver sites for a meteor velocity to be determined. The 8 km east-west separation of the receiver stations means that beams narrower than about 2° will only begin to overlap at a range of 200 km, or if skewed in azimuth, ensure an overlap for a limited range interval. The current beam width for the 12λ remote site receiver arrays is 4.3° ensuring overlap from a range of 100 km. Narrowing the receiver beams further would reduce the number of meteors detected on all three stations and hence the number of successful orbit calculations. Narrowing the Home site aerial further while still maintaining an overlap with the remote sites would be possible. The velocity measurement scheme of AMOR is based on the premise that the three echo profiles are measured under approximately similar conditions. Having radically different gains for the three receiver arrays would violate this and reduce the accuracy of the timelags.

The effective beamwidth can be narrowed by lengthening the transmitter array. Increasing the transmitter antenna to a broadside array 40λ long would narrow the transmitter beam to a width of about $\pm 0.7°$. Feeding all elements equally and in phase for an array of this size becomes correspondingly more difficult. To justify narrowing the beam in this way it is necessary to ensure that the needle of power is directed north-south to a precision of better than half a degree.

Narrowing the transmitter beam also has a practical limit. A sufficient length of the meteor trail must be illuminated for radar echoes to be detected at the remote stations. At 200 km the Home and remote site specular reflection points are separated by 4 km for a horizontal east-west ionisation column. Both these points need to be illuminated by the narrowed transmitter beam to detect the meteor from both sites. In fact very few meteors are detected in horizontal motion so the situation is not as extreme as this would suggest. Allowing for the variation in zenith angles of meteor velocities becomes a complex exercise in three dimensional geometry and empirically inspecting the components of the meteor velocities observed by AMOR is far easier. The average separation of reflection points in the horizontal east-west direction is 0.5 km with an extreme value of 1.6 km.[3] With a 1.4° angular beamwidth the

[3]Measuring the total separation between reflection points on the trail gives an average of

transmitter should illuminate a width of about 4.5 km at 200 km.

Measuring the angle of arrival for the meteor echo would increase the precision with which the azimuth angle to the meteor trail could be determined. By splitting one of the Home site receiver arrays into two components it would be possible to use this as an interferometer to measure the azimuth direction to the meteor trail. If the relative phase between the two sections can be measured to within 10°, as for the elevation phase comparison, then the 8λ separation between the two sections should provide angular resolution of about 0.2° in azimuth. This assumes that the physical alignment is accurate to within the same limits and would almost certainly require calibration, for example, by an aircraft towing a transmitter.

8.2.2 Higher Pulse Rate

The timelags between the occurrence of the echo amplitude profiles at each of the receiver sites are calculated to an accuracy of one pulse. The 379 Hz pulse rate used by AMOR gives a timelag accuracy of ± 2.6 ms. Increasing the pulse rate would give more data points up the rising edges of the echo profiles and may give a corresponding improvement in the timelag accuracies. I suspect that the accuracy of the timelag velocity method is dominated by the physics involved in the formation of the ionisation column and variations in the atmospheric conditions around each of the different reflection points rather than the ability to measure it sufficiently accurately. The numerical accuracy of about a tenth of a pulse found for the differential method used to locate the maximum rising edge slope was much smaller than the variation arising from individual profiles.

Increasing the pulse rate would give more frequent sampling of the Fresnel oscillations as the meteor particle moves away from the specular reflection point and would extend Fresnel velocity comparisons for the timelag velocity method to higher velocities. Such a comparison would be useful in an effort to clarify the reliability of observations that show meteoroids in hyperbolic orbits. More frequent echo amplitude measurements from an increased pulse rate would make determinations of the atmospheric deceleration values more accurate.

Range measurements from timing of the arrival of the meteor echo become more seriously ambiguous for pulse rates higher than that of the AMOR. A technique of doubling every n^{th} pulse has been used to resolve this ambiguity, for example Cook *et al* (1972) and this could be used although it would introduce some extra complexity in the data acquisition software. Meteor echoes from some ranges will occur just as a subsequent transmitter pulse is being sent and hence not be detected.

More frequent transmitter pulses would reduce the peak power available from the present transmitter which has an average power limit of 700 Watts.

2.5 km and an extreme value of 4.6 km.

Decreasing the pulse width while keeping the peak power constant would require wider bandwidth giving more chance of interference and reduced signal to noise ratio. If the pulse length from the receiver output becomes much shorter the frequency of amplitude sampling during a range scan would have to be increased to ensure detection of echoes at any range; see the comments below about more frequent A/D sampling.

8.2.3 More Frequent A/D Sampling

More frequent A/D sampling during a range scan would allow the range of a meteor trail to be determined with greater precision. The echo amplitude recorded for each radar sweep uses three adjacent rangebin samples. With closer sampling the echo amplitude recorded by AMOR will be less dependent on the relative position of the echo peak with respect to the rangebin sampling points. Increasing the amount of data recorded for each radar sweep would fill the computer memory more rapidly reducing the length of time available for extracting the observation records before the data is overwritten.

The 40 μs rangebin sampling was lengthened from the designed sample rate of 32 μs to give the daisy chain EPROM microprocessors sufficient time to complete their meteor detection routines. The computer still requires sufficient time during each range scan to reinitialise the DMA controller and run a detection routine on the previous sweeps data. An increased data acquisition rate also reduces the amount of spare time available for the computer to extract and assemble the meteor observation records. There may be sufficient time for the 12 MHz AT computer to use 32 μs rangebin sampling.

8.2.4 Improved Accuracy in Elevation Angle

The elevation angle of a meteor echo is determined by an interferometer at the Home site. The 5λ separation of the two receiver antennas gives an uncertainty of $\pm 1°$ for the elevation angle. Increasing the separation of the two Home site receiver aerials to 12λ would, assuming the current 10° uncertainty in the relative phase, would give an elevation accuracy of $\pm 0.33°$. Greater separation of the two antennas would improve accuracy in both the position of the meteor and the orientation of the plane in which it is moving. For a typical 200 km range the geometric height of the reflection point could be determined to within 1 km (providing a substantial improvement over the present 3.6 km accuracy for studies of meteor-atmosphere aeronomy).

Moving the receiver antennas to this separation would render the current solution to solving ambiguous phase to elevation conversions unmanageable. A third receiver array would have to be introduced to provide a coarse determination of the elevation angle. Providing a separation of 3λ and relative phase measurements between two antennas in conjunction with the echo range would give unambiguous low resolution elevation angles for all meteor trails

from the horizon to the local zenith.

The current configuration of AMOR using open wire transmission lines to bring the signals from the antennas to the receiver hut for phase comparison is sensitive to moisture levels (see section 4.3.4). Making consistently reliable phase sensitive measurements on three antenna arrays separated by 12λ would probably require phase sensitive receivers at each of the aerials (representing a significant increase in receiver technology) or else high quality coaxial transmission lines.

8.2.5 Broader Elevation Pattern

The AMOR beam is about 20° wide in elevation producing a fairly severe limit on the accessible radiant points. Replacing the transmitter antenna with a collinear array would broaden the beam in elevation from 20° to about 27°. Lowering the active elements in the aerial arrays from 0.6λ above ground to 0.4λ would place the beam elevation beam between 3 dB points at 18° and 71° with a maximum at about 40°. Timelag uncertainties for meteors detected below 18° substantially reduce the accuracy of the velocity measurements and very few meteors would ever be detected above 71°.

The broader beam would give a greater and more even coverage in declination. Searching within the AMOR data set for streams associated with Earth crossing asteroids would be less prone to observation biases. It would make it possible to observe most streams with southern ecliptic latitudes at any time during the year.

The problems of phase-elevation ambiguities encountered with the present AMOR configuration are compounded for higher elevation echoes. Unambiguous elevation angle determinations are necessary before the beam can be usefully broadened in elevation.

Widening the collecting area in elevation will reduce the antenna gains and hence magnitude limit achieved by AMOR. Most of the antenna gain is achieved in azimuth, the elevation gain factor is only two or three so the reduction in sensitivity would not be critical.

Meteor trails at higher elevations are correspondingly closer. The overlap between the individual beams of the spaced receiver stations drops as the ground plane range to the target trail gets shorter. The number of meteors with observations from all three sites would decrease. Since the meteor is moving in a plane more closely parallel to the ground plane the accuracy of velocities calculated from the timelags is improved. For higher elevation meteors a smaller baseline separation of the receiver sites still gives an equivalent accuracy. Adding in two extra closer remote receiver sites would help overcome this reduction in the beam overlaps at higher elevations.

8.2.6 Continuous Operation - Greater Security

The Birdlings Flat field station has been burgled eight times during the period of my PhD research. A major factor in the design of the current AMOR system was the need to make it portable so that when not in operation it could easily be returned to the University campus. The five burglaries in the first two years caused a delay of about two months each while some item of custom built equipment was rebuilt. The current configuration of AMOR is capable of continuous operation and on site data reduction. We are too worried about equipment being stolen while the field station is unattended to do this.

The extra observing time would be invaluable in searching for southern hemisphere meteor streams. Having great gaps because the computer has been stolen would tend to negate any gains from continuous operation! Leaving the equipment at the field station would reduce the damage caused by transportation. The battery power supply for the remote sites provides about five days of unattended operation. It would be relatively easy to extend this by adding more batteries.

8.2.7 More Remote Stations

Extra remote stations would increase the number of successful meteor orbits by making it more likely that each meteor was observed from at least three receiver sites. For the few meteors with good Fresnel patterns an extra echo profile would allow greater precision in measuring atmospheric decelerations. Having an extra station close to the Home site would improve the detection of meteors at high elevations. An extra station would provide an inherent internal check on the velocity measurement and improve the automated rejection of echo profiles not observed until some time after the formation of the ionisation trail.

Any increase in the amount of data collected by the system will put corresponding timing pressures on the meteor detection and recording software within the computer.

8.2.8 Beam to North and South

The rhombic transmitter array was designed to direct most power into the south lobe and hence implicitly locate meteor trails to the south.[4] Timelags can be used to determine whether a meteor is moving south or north and hence determine which lobe the meteor trail is in. The AMOR system in its current configuration does detect meteors in the north and collects sufficient information to calculate the polar orbits of these particles. Changing the transmitter array to a collinear configuration would produce two equal lobes pointing south and north. More meteors in polar orbits would be detected

[4]The rhombic array has been the most reliable component of the AMOR system. Only now once everything else is starting to work well would I dare to suggest changing it.

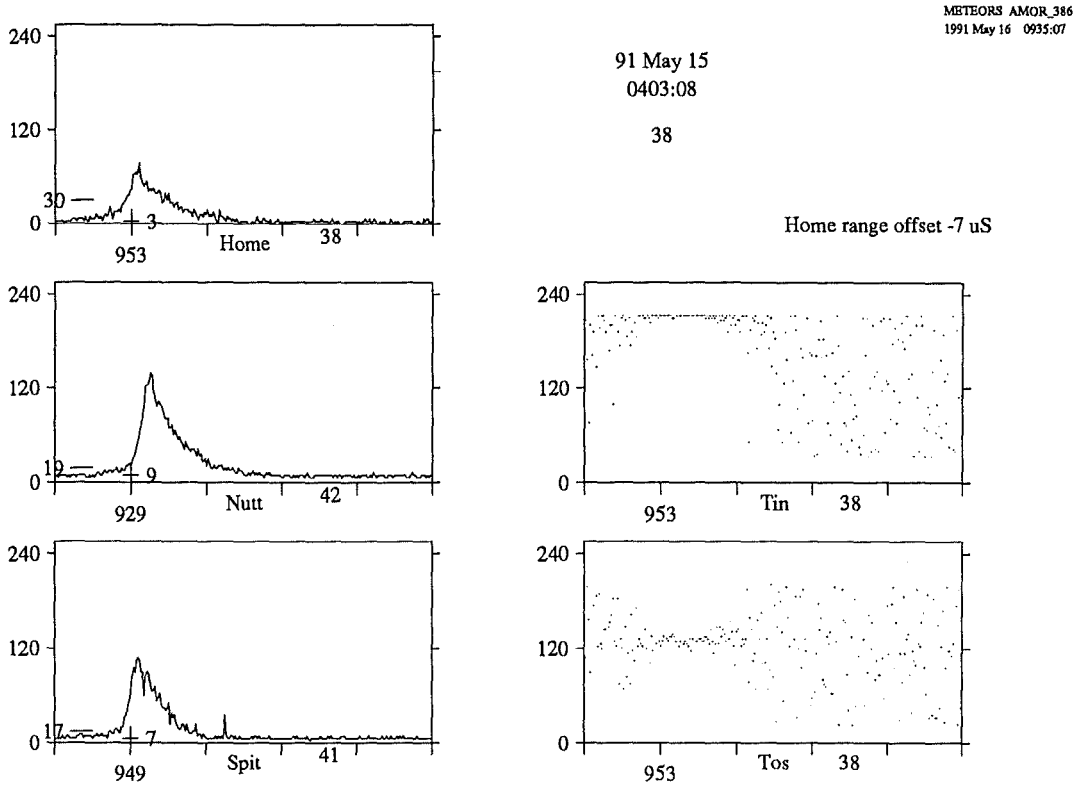
and the detection probability would then be the same as those in the ecliptic making studies of their relative density more straight forward.

Putting half the power into a north lobe would reduce the gain of the transmitter array. However this would be partially balanced by not dissipating any power in the resistor loads on the rhombic. Increasing the length of the antenna would provide a compensating increase in the gain of the transmitter array. The rhombic loads need to be periodically checked and replaced as the component resistors corrode in the sea air. How a long broadside collinear array will respond to transmitter power levels remains to be seen.

8.2.9 UHF Data links

The lack of commercial FM data links with a sufficiently large bandwidth dictated the original selection of 39 MHz for the communication links from the remote receiver stations. It has required a considerable effort to bring these custom built FM links to reliable operation.[5] Upgrading to a 450 MHz FM link system based on a commercial transmitter receiver combination is currently under investigation. The smaller physical size of link antennas for this frequency would make them easier to maintain and their more highly directional properties would almost certainly reduce interference from external sources and hopefully eliminate the worst of the corona emissions from the local 11 kV power transformer for the field station.

[5] Most recently the upgrade to 40 amp.hour batteries has enabled the remote site electronics to remain connected and in operation for days at a time making a a big difference to reliability of the link. The consequent reduction in transportation of the links must be a factor in this improvement.



91 May 15 0403:08 38 Reduced Data

	Noise	Range	Max	Rise	Imax	Decay	DiffHeight	Altitude	Fres Vel	1/2 Cycles
Home	3	204	67	11	54	26	97.1	99.9		
Nutt	9	212	130	10	63	27	96.8	101.9		
Spit	7	208	102	10	55	18	99.7	100.3		

	Lag12	Lag13	Lag23					
Timelags	15.3	3.1	-12.2	Tin	213	Phase	Elevation	
				Tos	130	129	28.5	

Observed Velocity

Azimuth angle	16.4	Vx	-26.8	Persistent (s)	0
Zenith angle	29.6	Vy	-7.9	Fresnel Vel	0.0
Speed	56.6	Vz	-49.2	Atmos Decel	0.0

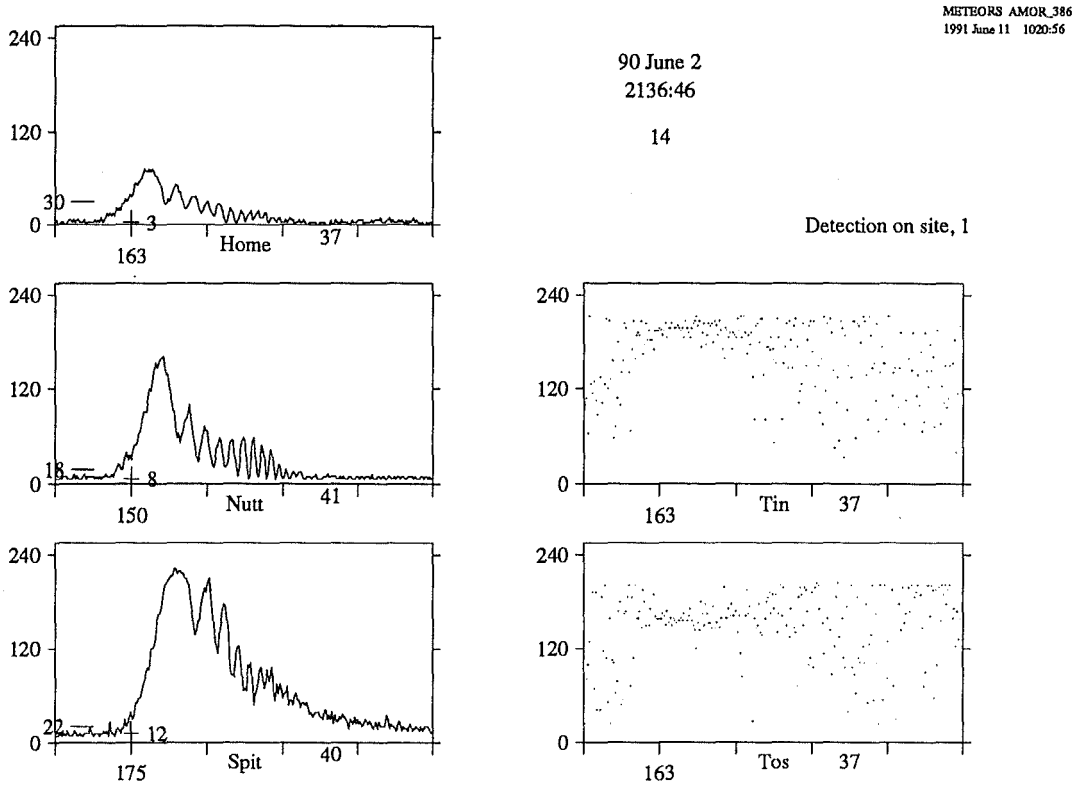
Heliocentric Position and Velocity

				Corrected Geocentric Radiant		
Day Number	134	Vmx	-9.6	Right Asc	Declination	Speed Geo
Orbital radius	1.011	Vmy	40.3	278.3	-14.9	57.0
Speed Helio	42.2	Vmz	-8.3			

Orbital Elements

Peri Dist	Ecc	Inclin	Arg Peri	Long Asc 1950	Semi Major	-29.84
					Peri Lat.	-0.39
0.09	1.00	139.20	-35.44	53.75	Peri Long.	1.43

Figure 8.8: The meteor observation and reduced data for an individual meteoroid orbit summarises the operation of AMOR.



90 June 2 2136:46 14 Reduced Data

	Noise	Range	Max	Rise	Imax	Decay	DiffHeight	Altitude	Fres Vel	1/2 Cycles
Home	3	197	71	21	61			95.2	18.6	16
Nutt	8	206	156	18	69			95.5	18.8	20
Spit	12	202	220	21	80			93.9	19.2	13

	Lag12	Lag13	Lag23		Tin	200	Phase	Elevation
Timelags	6.4	-25.2	-31.6		Tos	155	135	28.1

Observed Velocity

Azimuth angle	322.4	Vx	-9.9	Persistent (s)	0
Zenith angle	34.0	Vy	7.7	Fresnel Vel	18.9
Speed	22.4	Vz	-18.6	Atmos Decel	4.8

Heliocentric Position and Velocity

Day Number	153	Vmx	36.6
Orbital radius	1.014	Vmy	18.9
Speed Helio	41.2	Vmz	-2.0

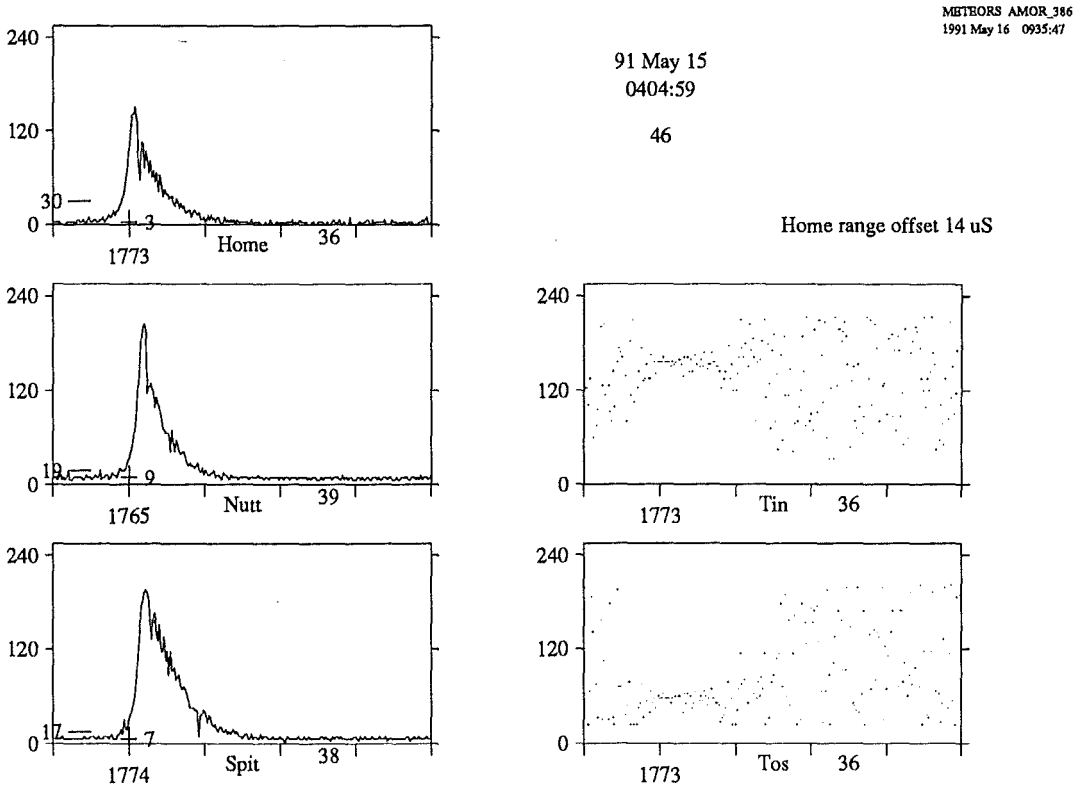
Corrected Geocentric Radiant

Right Asc	Declination	Speed Geo
230.0	-12.7	20.6

Orbital Elements

Peri Dist	Ecc	Inclin	Arg Peri	Long Asc 1950	Semi Major	17.63
0.80	0.95	3.11	-124.00	71.03	Peri Lat.	-0.04
					Peri Long.	5.36

Figure 8.9: Meteor observation showing strong Fresnel diffraction effects for which an atmospheric deceleration has been determined.



91 May 15 0404:59 46 Reduced Data

	Noise	Range	Max	Rise	Imax	Decay	DiffHeight	Altitude	Fres Vel	1/2 Cycles
Home	3	189	133	9	53	19	99.3	104.0		
Nutt	9	194	186	9	59	21	98.6	104.2		
Spit	7	190	188	9	61	22	98.3	102.6		

	Lag12	Lag13	Lag23					
Timelags	1.6	-8.5	-10.1	Tin	156	Phase	Elevation	
				Tos	57	63	32.7	

Observed Velocity

Azimuth angle	320.6	Vx	-38.8	Persistent (s)	0
Zenith angle	39.7	Vy	31.9	Fresnel Vel	0.0
Speed	78.6	Vz	-60.5	Atmos Decel	0.0

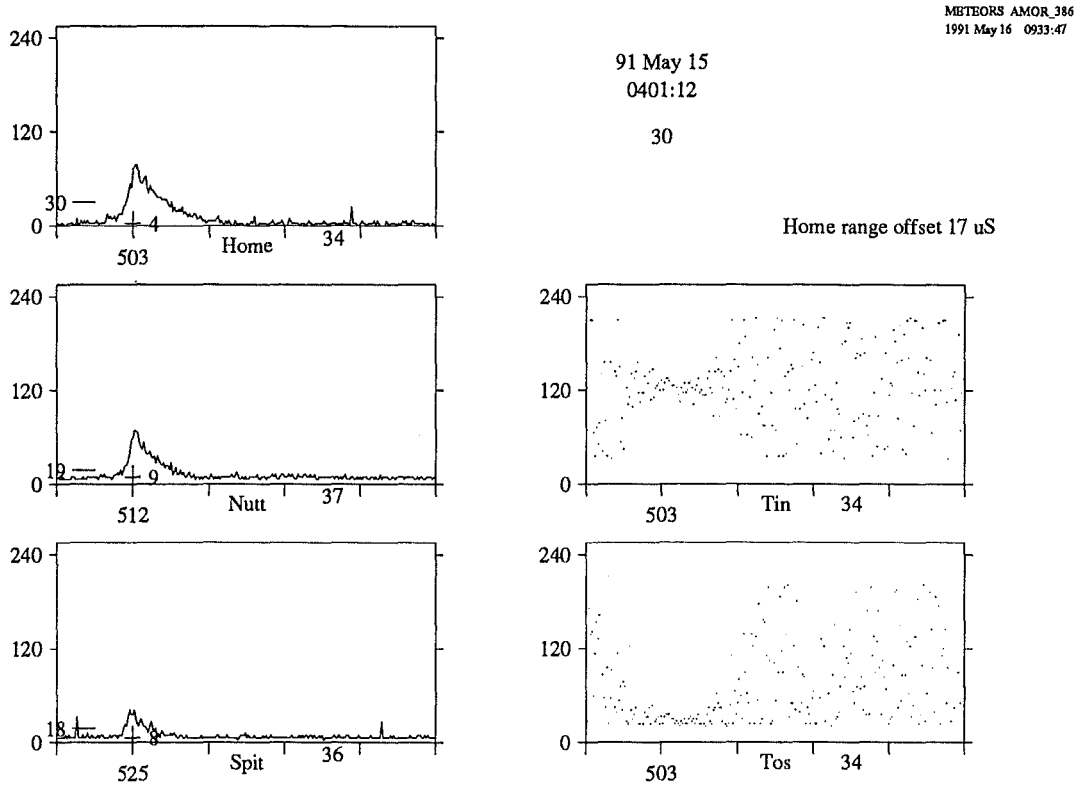
Heliocentric Position and Velocity

				Corrected Geocentric Radiant		
Day Number	134	Vmx	-47.5	Right Asc	Declination	Speed Geo
Orbital radius	1.011	Vmy	17.6	311.5	-10.0	79.8
Speed Helio	51.8	Vmz	-10.7			

Orbital Elements

Peri Dist	Ecc	Inclin	Arg Peri	Long Asc 1950	Semi Major	-0.95
0.92	1.97	167.28	-150.17	53.75	Peri Lat.	-0.11
					Peri Long.	3.57

Figure 8.10: Meteor observation for a particle in a hyperbolic orbit.



91 May 15 0401:12 30 Reduced Data

	Noise	Range	Max	Rise	Imax	Decay	DiffHeight	Altitude	Fres Vel	1/2 Cycles
Home	4	176	71	10	52	20	99.0	100.6		
Nutt	9	182	64	9	51	22	98.3	99.9		
Spit	8	178	37	8	49	16	100.6	98.9		

	Lag12	Lag13	Lag23		Tin	124	Phase	Elevation
Timelags	-9.1	-20.4	-11.2		Tos	29	36	34.2

Observed Velocity

Azimuth angle	296.4	Vx	-21.8	Persistent (s)	0
Zenith angle	56.9	Vy	43.9	Fresnel Vel	0.0
Speed	58.5	Vz	-32.0	Atmos Decel	0.0

Heliocentric Position and Velocity

				Corrected Geocentric Radiant		
Day Number	134	Vmx	-28.6	Right Asc	Declination	Speed Geo
Orbital radius	1.011	Vmy	-11.4	335.5	-5.9	59.2
Speed Helio	31.1	Vmz	-4.2			

Orbital Elements

Peri Dist	Ecc	Inclin	Arg Peri	Long Asc 1950	Semi Major	1.12
0.70	0.38	171.70	82.56	53.75	Peri Lat.	0.14
					Peri Long.	-0.50

Figure 8.11: Small amplitude short duration meteors still give reliable orbits using the maximum rising-edge slope method.

8.3 Summary

A Meteor Orbit Radar (AMOR) has been constructed at Birdlings Flat near Christchurch, New Zealand. The AMOR system uses a narrow beam pulsed radar to locate the meteor with velocities determined by timing the difference between the onset of echo detections at three spaced receiver stations. Computerised data acquisition, automated reduction to orbital elements and interactive graphical display are a significant feature of the system.

Figure 8.8 encapsulates the essence of AMOR observations. The top half of the display includes all the information recorded at the time of observation. It includes 250 echo amplitude samples at 2.64 ms intervals from each of the three receiver stations. Two profiles with phase information are recorded from the Home site interferometer. Timelags between the location of the steepest rising-edge slope on each of the station profiles gives the observed velocity. Average values from the Tin and Tos profiles are used to calculate the elevation angle of the meteor trail. Corrections for having observed the meteor from on the Earth and coordinate transformations to a heliocentric reference frame are implemented using a vector based notation rather than spherical trigonometry. Some of the information from intermediate steps in this reduction is recorded in the reduced data records and displayed in the bottom half of Figure 8.8. Finally the orbital elements for the meteoroid orbit are included.

The AMOR system uses a 26.2 MHz pulsed transmitter with a 20 kW peak power output. 66 μs pulses are transmitted every 2.64 ms, giving a pulse rate frequency of 379 Hz. Long antenna arrays give a narrow beam in azimuth with gain factors of 427 and 133 for the transmitter and Home receiver arrays respectively making detection of meteors with a radar magnitude of +12.5 possible. The narrow beam locates the meteor trail to within $\pm 1.4°$ of directly south or north of the radar station. Whether the trail is to the north or south is resolved by requiring specular reflection and noting the relative motion of the meteor. The range to the trail is calculated from the time delay between the transmission of a pulse and detection of the echo. Sampling in 40 μs rangebins gives a range resolution of 6 km. Fitting the receiver output profile on the three rangebin amplitudes centred on the echo gives an offset from which an accuracy of 1 km for the Home site range can be achieved. The elevation angle to the meteor is determined by measuring the relative phase angle of the received echo between two antennas separated by 5λ. The measured phase difference is calibrated by requiring a peak in the distribution of meteor heights between 95 and 99 km. Ambiguities in converting from the relative phase angle to an elevation angle are resolved by assuming the meteor trail occurs in the height band from 70 to 120 km. The range, elevation and narrow beam fix the position of the meteor.

To obtain sufficient information about the meteor particle to calculate its orbit AMOR needs to measure its velocity. Specular reflection from a long thin ionisation trail requires that the meteor which formed it be moving in a plane

perpendicular to the vector from the radar station to the meteor trail. The elevation angle and narrow beam fix this plane. The reflection points on the ionisation trail for three spaced receivers will, in general, be different. As the meteor moves through each of the reflection points the ionisation formed there will cause the received echo amplitude to increase. By measuring the relative timelags between the rising edge of the meteor profile at each of the three receiver stations the velocity in the plane of motion can be calculated. The timelags are determined by locating the occurrence of the maximum slope in the rising edge and are accurate to ± 1 radar pulse ($\pm$ 2.64 ms). This maximum rising-edge slope timelag scheme was chosen as being least affected by differences in atmospheric condition for the three different reflection points. Effectively the two components of the meteor velocity in the ground plane are measured and the meteor speed determined by projected this up into its plane of motion. Specular reflection and timelags between three spaced receivers determine the velocity vector for the meteor.

Previous orbit radar systems have used Fresnel diffraction effects to determine the speed of the meteor particle. Calculations of Fresnel based velocities are attempted for all observation profiles by the automated AMOR data reduction program. Figure 8.9 shows an observation with clear Fresnel diffraction effects. Comparison with the timelag based velocities for for the 1.6×10^4 observations (up to 1991 May 16) on which Fresnel velocities could be determined showed a complete agreement. Estimates of the experimental uncertainties from this comparison suggest an accuracy of ± 2.5 km/s for AMOR velocities. The great advantage of the new timelag method employed by AMOR for an orbit radar is evident by comparing the number of Fresnel based velocities with the 1.3×10^5 successful timelag determinations in the same period. Because of pulse rate limitations the Fresnel based method is not well suited to observing especially high velocity meteors. Figure 8.10 shows the meteor observation for a particle in a hyperbolic orbit that did not have sufficient information for determining a Fresnel velocity.

The Fresnel based calculation scheme used within the AMOR package uses all points in the echo amplitude profile to fit the velocity. Calculating velocities for subdomains of the oscillation profile makes it possible to determine the atmospheric deceleration of the particle as it moves through successive Fresnel zones. To ensure reliability of the results only those observations with two or three overlapping velocity sequences are used in calculating atmospheric decelerations. Values between -5 and 40 $km.s^{-2}$ are typical although atmospheric decelerations as high as 100 $km.s^{-2}$ have been measured.

Observing meteors provides information from which the orbital distribution of of the interplanetary dust complex can be probed. Before the meteoroid orbit can be determined the velocity of the particle before it encountered the Earth needs to be calculated. The AMOR system handles corrections for the presence of the Earth using a new vector based scheme which is described in Chapter 6 and summarised in Table 8.2. The vector notation greatly simplifies

some of the steps in comparison with previous derivations and allows a more appropriate order for the correction process. That the quadrant information is implicitly contained within the velocity vector is a distinct advantage in the computer implementation of this approach. Having the components of the heliocentric velocity stored in the reduced data record greatly simplifies the conceptual problems of identifying the orbit as prograde or retrograde and whether, at detection, the node is ascending or descending and the motion toward or away from perihelion.

AMOR was used to observe the 1990 η Aquarid meteor shower as a complete astronomical test for system. Stream meteors have been identified by two different methods, the simpler based on radiant position and heliocentric velocity and the second a direct comparison of the orbits using the modified Drummond D-Criteria. Requiring the radiant to be within a small box aligned to the daily motion of the stream radiant combined with a heliocentric velocity between 34 and 48 km.s^{-1} gave 361 stream members with a mean orbit in good agreement with previous determinations. The second selection method required each member to be within a D-Criteria range of 0.04 with at least two other members. This search was seeded using the mean orbit found from the first method and obtained a less contaminated sample of 270 η Aquarid meteor orbits. The mean orbital elements for these D-Criteria stream meteors are included in Table 8.3 along with the standard deviation of the sample and the standard error of the mean. The elements quoted for the 1990 AMOR stream orbit give the peak or modal values of the orbital element distributions. An error analysis for a typical shower meteor observation using uncertainties of timelag $\pm$ 1 pulse, elevation angle $\pm$ 0.5° and azimuth direction to trail $\pm$ 1.0° was conducted. The extreme uncertainty limits for the orbital elements of an individual η Aquarid meteor orbit that this implies are tabulated in Table 8.3. Observations of stream meteors can only be taken while the shower radiant is in the sky so the mean value for the longitude of the ascending node cannot really be determined to an accurately better than one degree for a single years observing and depends heavily on the actual hours spent observing on each day.

The η Aquarid shower members show a shift in the daily mean radiant point indicating an orbital stream observed at successive longitudes of the descending node. This is the first time this daily shift in the radiant has been clearly demonstrated with radar observations. A series of orbit density cross sections show a pronounced enhancement of the meteoroid orbits around the mean stream orbit within the AMOR data set. These cross sections were taken by making small variations in each of the three components of the heliocentric velocity about the central stream values and counting the number of observed orbits within the specified D-Criteria range. Comparing the number of meteors in a similar orbit before and after the shower shows a marked increase in stream meteors between 1990 April 29 and May 12 confirming its period of activity.

- Observed Meteor Velocity $\mathbf{v}_1 = -v_1 \begin{pmatrix} \sin z \cos A \\ \sin z \sin A \\ \cos z \end{pmatrix}$
  - Observed Speed, v_1
  - Zenith angle, z
  - Azimuth angle of velocity, A
- Atmospheric Deceleration $v_2{}^2 = v_1{}^2 + 0.80 v_1{}^{1.6}$
  - Meteor Ablation Theory
  - Fresnel Measurements
- Rotation Velocity of the Earth $\mathbf{v}_3 = \mathbf{v}_2 - \mathbf{v}_{\text{Rotation}}$
  - Velocity of Radar Station w.r.t. Centre of Earth
- Gravitational Acceleration $\frac{1}{2} m v_3{}^2 - \frac{\mathrm{GM}_\oplus m}{R_m} = \frac{1}{2} m v_4{}^2$
  - Conservation of Energy
- Zenith Attraction $\tan\left(\frac{1}{2}\triangle z\right) = \frac{v_3 - v_4}{v_3 + v_4} \tan\left(\frac{1}{2} z'\right)$
  - Hyperbolic Geometry
- Equatorial Coordinates $\mathbf{v}_5 = \mathbf{A_2 T A_1 v}_4$
  - Latitude of Station, $\mathbf{A}_1$ and $\mathbf{T}$
  - Mean Sidereal Time (BFMST - $\pi/2$), $\mathbf{A}_2$
- Heliocentric Ecliptic Coordinates $\mathbf{v}_6 = \mathbf{A_4 A_3 v}_5$
  - Earth's Inclination, $\mathbf{A}_3$
  - Solar Longitude (LS - $\pi/2$), $\mathbf{A}_4$
- Orbital Motion of the Earth $\mathbf{v}_m = \mathbf{v}_6 + \mathbf{v}_\oplus$
  - Deviation of Apex

Table 8.2: The corrections necessary to convert an observed meteor velocity, $\mathbf{v}_1$, to a heliocentric velocity, $\mathbf{v}_m$, of the particle before it encountered the Earth are summarised from Chapter 6.

Summary of AMOR η Aquarid orbital elements

	q	e	i	w	Ω_{1950}	V_m	Number
Stream Orbit, AMOR Christchurch 1990 (modal values).							
	0.57	0.98	165.5	97	46	41.5	270
1990 D-Criteria stream members,					DD < 0.04,		2 sponsors.
	0.555	0.983	165.5	94.5	45.5	41.3	270
s.e.	0.004	0.008	0.1	0.7	0.2	0.2	
s.d.	0.070	0.138	2.2	11.2	3.7	2.8	
Extreme uncertainty limits for individual orbits.							
	0.33	0.76	163.0	62		35.6	
	0.77	1.45	167.5	126		48.5	
1991 D-Criteria stream members,					DD < 0.04,		2 sponsors.
	0.560	1.013	166.0	96.0	45.2	42.0	257
s.e.	0.002	0.009	0.1	0.6	0.2	0.2	
s.d.	0.063	0.141	2.3	9.6	3.9	2.7	

Table 8.3: A Summary of the orbital elements found by AMOR for the η Aquarid stream. These include the mean elements calculated for the 1991 apparition of the shower.

The orbital elements found for the stream closely match those of the Comet Halley apparitions between 607 AD and 989 AD.

The performance and utility of the AMOR system is most graphically demonstrated in the handling of the 1991 η Aquarid observations. The system ran continuously from 1991 April 27 to May 16. All 25 206 meteor orbits for this period were completely reduced by May 17 and 257 η Aquarid stream meteors identified. Figure 8.12 radiants for these are compared with all η Aquarid orbits determined up to 1988 and summarised by Lindblad (1989). The mean orbital characteristics for the stream agree well with the data from the 1990 apparition used to calibrate AMOR. Figure 8.12 is important in demonstrating the very large improvement in meteor stream characteristics that can be achieved by AMOR.

As part of my PhD research I oversaw the production of AMOR developing many aspects of hardware, all the software support and data reduction, including the display routines used to produce all the graphs in this thesis as well as those diagrams written directly in native postscript. Bringing AMOR to an operational status included a full test of the system for astronomical observation on the η Aquarid meteor stream. Many programs were also developed for more general studies of meteoroid dynamics. A total of 1.3×10^5 meteor orbits have been produced in the 128 days of observing over the 14 months preceding 1991 May 16.

8.4 Conclusion

I built a meteor orbit radar. It works.

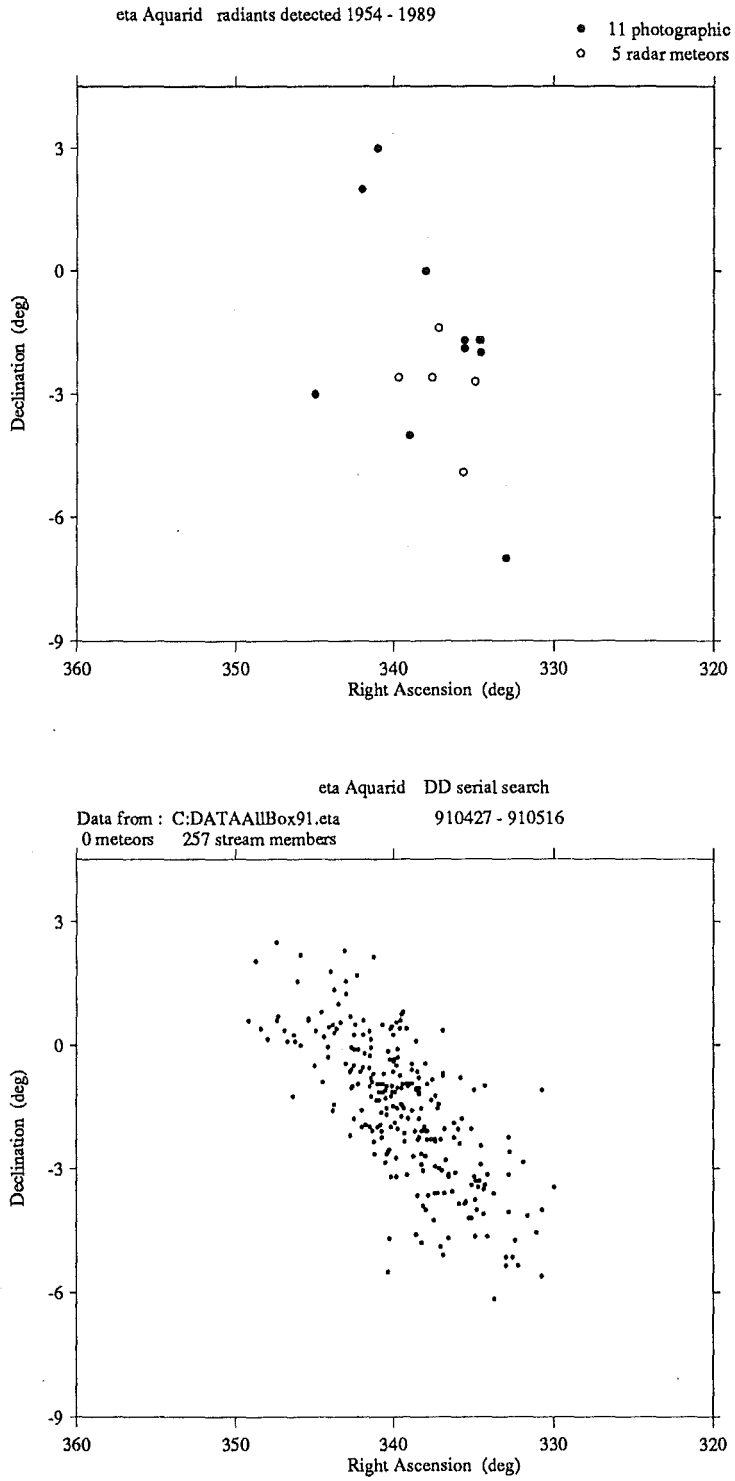


Figure 8.12: An example of the substantial improvement achieved by AMOR. The top panel graphs the radiant points of the 16 η Aquarid meteor orbits determined up to 1989 summarised by Lindblad (1989). The second panel presents data collected by the Christchurch AMOR system during the period 1991 April 27 to May 16.

Acknowledgements

The AMOR project is very much a team effort, my role being more that of coordinating other peoples technical expertise and developing the necessary software. My thanks to Professor Jack Baggaley whose enthusiasm for getting things started and ability to keep my research objectives realistic has made this thesis possible. His wide knowledge in the field has provided a firm reference for me to work within.

It has been a privilege to work with Bob Bennett who would regularly distract me with a story while solving the current crisis. My thanks go to Graham Lees whose help and assistance make him an invaluable workmate. The production line construction of receiver aerial arrays and his tireless efforts to keep the field station and AMOR running are appreciated. My thanks to the workshop staff, especially Ross Ritchie and Geoff Graham, who continued to remain friendly each time I came up with another new idea. To Duncan Steel, my unseen co-worker and predecessor, thank you for bringing the radar within a PhD of operation.

I am proud of the Physics Department that I have worked in over the last four years. My personal thanks for the support and encouragement of all those people in it. Particular thanks go to my fellow students of a passing generation, Bill Brown and Bryan Lawrence for always being around to help. The members of the Union Rowing Club who have given me a life away from University are an important part of this work.

To my parents who supported me through my PhD and in the end financed it as well, this is yours too. For Jane who insisted she should understand most of the introduction and Anthony who led by example, thankyou.

References

Almanac for Computers. (1989). Nautical Almanac Office, US Naval Observatory, Washington DC.

Astronomical Almanac. (1990). Nautical Almanac Office, US Government Printing Office, Washington DC.

Allen C. W. (1973). *Astrophysical Quantities.* Althone Press, London.

Andrianov K. N. S., Pupysev U. A. and Sidorov V. V. (1970). The distribution of orbit parameters and the changes in incident meteor particle flux density. *Mon. Not. R. Astr. Soc.* **148**:227 – 237.

Beltlam and de Lignie. (1989). Multistation meteor photography in the Netherlands. *Third International Symposium on Asteroids, Comets, Meteors.* Lagerkvist C.-I., Magnusson P., Rickmann H. (eds), Uppsala, Sweden.

Bennett R. G. T. (1953). *On the Measurement of Meteoric Velocities by the Radio Method.* MSc Thesis, Canterbury University College, University of New Zealand.

Bronshten V. A. (1983). *Physics of Meteoric Phenomena.* D. Reidel Pub. 356 pages.

Bruce E., Beck A. C. and Lowry L. (1935). *Proc. Inst. Radio Engin.* **23**:24.

Burns J. A., Lamy P. H. and Soter S. (1979). Radiative forces on small particles in the solar system. *Icarus* **40**:1 – 48.

Ceplecha Z. (1988). Earth's influx of different populations of sporadic meteoroids from photographic and television data. *Bull. Astron.Inst. Czechosl.* **39**:221 – 236.

Ceplecha Z. (1990). *IAU Commision 22, Report 1990.*

Clegg J. A. (1948). Determination of meteor radiants by observation of radio echoes from meteor trails. *Phil. Mag.* **39**:577-594.

Comrie L. J. (1950). *Mathematical Tables.* W and R Chambers, London.

Cook A. F., Flannery M. R., Levy II H., McCrosk R. E., Sekanina Z., Shao C.-Y., Southworth R. B. and Willians J. T. (1972). *Meteor Research Program* NASA CR-2109, Smithsonian Astrophysical Observatory, Washington DC,

Cook A. F. (1972). A working list of meteor streams. *Meteor Research Program* NASA CR-2109, Smithsonian Astrophysical Observatory, Washington DC,

Denning W. F. (1899). General catalogue of the radiant points of meteoric showers and of fireballs and shooting stars observed at more than one station. *Memoirs R. Astr. Soc.* **53**:203.

Davies and Gill (1960). Radio echo measurements of the orbits of faint sporadic meteors. *Mon. Not. R. Astr. Soc.* **121**:437 – 462.

Drummond J. D. (1981). A test of comet and meteor shower associations. *Icarus* **45**:545 – 553.

Drummond J. D. (1979). On the meteor/comet orbital discriminent D. *Proc. Southwest Reg. Conf. Astron. Astrophys.* **5**:83 – 86.

Eggbrecht L. C. (1987). *Interfacing to the IBM PC* Indianapolis.

Elkin W. J. (1899). *Astrophys. J.* **9**:20.

Ellyett C. D. and Davies J. G. (1948). Velocity of meteors measured by diffraction of radio waves from trails during formation. *Nature* **161**:596 – 597.

Fisher W.J. and Olmsted M. (1929). *Bull. Harv. Coll. Obs no870.*

Gartrell and Elford (1975). Southern hemisphere meteor stream determinations. *Aust. J. Phys.* **28**:591 – 620.

Gill and Davies (1956). A radio echo method of meteor orbit determination. *Mon. Not. R. Astr. Soc.* **116**:105 – 113.

GreenHow J. S. and Neufeld E. L. (1955). The diffusion of meteor trails in the upper atmosphere. *J. Atmos. Phys.* **6**:133 – 140.

Halliday I., Griffin A. A., Blackwell A. T. (1983). *Highlights of Astronomy* **6**:399 – 404.

Halliday I., Blackwell A. T. and Griffin A.-A. (1989). Detailed records of many unrecovered meteorites in Western Canada for which further searches are recommended. *J. R. Astron. Soc Can.* **83** no 2:49 – 80.

Hawkes R. L., Jones J. and Ceplecha Z. (1984). The populations and orbits of Double-Station TV Meteors *Bull. Astron. Instit. Czech.* **35**:46 – 64.

Hawkins (1964). *Meteors, Comets and Meteorites.* McGraw-Hill, New York. 134 pages.

HP Vectra Technical Reference Manual **1:Hardware** Hewlett Packard.

Hey J. S. (1949). Radar observations of meteors. *Radio Astronomy.* page 104.

Hey J. S. and Stewart G. S. (1947). Radar observations of meteors. *Proc. Phys. Soc.* **59**:858 – 883.

Hoffmeister C. (1948). Verlag Werden und Werken Weimar, Leipzig. *Meteorströme.* : in German chapter summaries in English.

Howick E. F. (1991). *Atmospheric Parameters from Three-Station Radio-Meteor Data.* MSc Thesis, University of Canterbury, Christchurch, New Zealand.

Jones W. and Jones J. (1990). Ionic diffusion in meteor trains. *J. Atmos. Terr. Phys.* **52**:185.

Jones J. and Sarma T. (1985). Double Station Observations of 454 TV Meteors *Bull. Astron. Instit. Czech.* **36**:103 – 15.

Kashcheqev B. L. and Lebedinec V. N. (1960). Radar studies of meteors. *Smithsonian Contributions to Astrophysics* **11**:183 – 190.

Kashcheqev B. L., Lebedinec V. N. and Lagutin M. F. (1961). Radar determination of the orbits of individual meteors. *Astron. J. USSR,* **38**:681 – 691.

Koseki M. (1989). Amateur meteor studies in Japan. *Third International Symposium on Asteroids, Comets, Meteors.* Lagerkvist C.-I., Magnusson P., Rickmann H. (eds), Uppsala, Sweden. (pages 543 – 546).

Kwa E. (1977). *The Effect of Ocean Tides on Antennae Impedence.* MSc Thesis, University of Canterbury, Christchurch, New Zealand.

Lebedinec V. N. (1968). Radar meteor orbits. *Physics and Dynamics of Meteors.* Kresak and Millman (eds), IAU Symposium **33**:241 – 264.

Lindblad B. A. (1971). Two computerised stream searches among meteor orbits. *Smithsonian Contributions to Astrophysics* **12**:14.

Lindblad B. A. (1989). The orbit of the Eta Aquarid meteor stream. *Third International Symposium on Asteroids, Comets, Meteors.* Lagerkvist C.-I., Magnusson P., Rickmann H. (editors), Uppsala, Sweden.

Lovell A. C. B. (1954). *Meteor Astronomy.* Clarendon, Oxford.

McIntosh, B. and Hajduk, A. (1983). Comet Halley meteor stream: a new model. *Mon. Not. R. astr. Soc.* **205**:931 – 943.

McIntosh, R. A. (1934). Ephemeris of the radiant point of the Eta Aquarid meteor stream. *Mon. Not. R. Astr. Soc.* **95**:601.

McKinley (1961). *Meteor science and Engineering.* McGraw-Hill, New York. 309 pages.

McKinley D. W. R. and Millman P. M. (1949). Determination of the elements of meteor paths from radar observations. *Canad. J. Res.* **A27**:53 – 67.

Manning L. A., Villard O. G. and Peterson A. M. (1949). Radio Doppler investigation of meteoric heights and velocities. *J. Appl. Phys.* **20**:475 – 479.

Millman P.M. and Hoffleit D. (1937). A study of meteor photograhs taken through a rotating shutter. *Ann. Harv. Coll. Obs* **105**:601 – 621.

Nilsson (1964). A southern hemisphere radio survey of meteor streams. *Aust. J. Phys.* **17**:205 – 256.

Olivier C. P. (1937). *Astron. J.* **46**:41.

Olsson-Steel D. (1987). Theoretical meteor radiants of Earth-approaching asteroids and comets. *Aust. J. Astron.* **2**:21 – 35.

Öpik E. J. (1948). *The Observatory* /bf 68:228.

Pecina P. (1988). Derivation of Fresnel characteristics with deceleration of the meteoroid taken into account. *Bull. Astron. Instit. Czech.* **39**:193 – 208.

Porter J. G. (1952). *Comets and Meteor Streams.* Chapman and Hall, London. 123 pages.

Prentice J. P. M. (1945). *British Astron. Assoc. Handbook* **36**

Rice D. and Forsyth P. A. (1963). Variations in meteoric radio signal decay rates. *Can. J. Phys.* **41**:679 – 690.

Schaiparelli J. V. (1866). *Stamferia.* Firenze: in Italian.

Shapley H and Opik E. J. and Boothroy S. L. (1932). *Proc. Nat. Acad. Sci. Wash.* **16**:16.

Simek M. (1973). Radio observations of the Geminids 1959-69. Overdense echoes. *Bull. Astr. Inst. Czech.* **24**:213.

Southworth R. B. (1972). Theoretical Fresnel patterns of radio meteors. *Meteor Research Program* NASA CR-2109:51 – 68.

Southworth R. B. and Hawkins E. S. (1963). Statistics of meteor streams. *Smithsonian Contributions to Astrophysics* **7**:261 – 285.

Verniani F. (1967). Meteor masses and luminosities. *Smithsonian Contributions to Astrophysics* **10**:181 – 195.

Verniani F. (1973). An analysis of the physical parameters of 5759 faint radio meteors. *J. Geophys. Res.* **78**:8429 – 8462.

von Niessel G. (1878). *Astron. Nachr.* **93**

Whipple F. L. (1951). Photographic meteor orbits and their distribution in space. *Astron. J.* **59**:201 – 217.

Yeomans, D. K. and Kiang, T. (1981). The long term motion of comet Halley. *Mon. Not. R. astr. Soc.* **197**:633 – 646.

Appendix A

Program File Descriptions

This section is a list of the programs associated with the meteor orbit radar. These notes consist of the program headers and internal descriptions for each source code file. They are arranged in groups with common purposes. The first two pages give a contents for the notes. This is followed by a program index to help find individual program descriptions.

Contents

A.1 Program Index

A.2 Observing

These programs are associated with using the computer during an observing run. Observe runs the radar. The other programs provide quick checks on the stored observation data. LookDisk has more general utility and is described under Observation Display Routines (Section A.7.2).

A.2.1 Main Observing Programs

```
program Observe;              {Meteor Orbit Observation Program.}
    uses  OrbLift, OrbRadar, OrbDefns, GenUtil, Crt, Dos;
{27 October  1989       Andrew Taylor.}
{This program operates the meteor orbit radar during an observing
 run.  It displays basic information about the run on the screen.
 A graphical display of the observation records stored on disk can
 be obtained by running the program LookDisk.  The program clicks
 every time it detects a new meteor echo on the Home site channel.
 Data overrun conditions and any runtime errors cause the program
 to terminate.  The exit routines set off a high pitched alarm and
 save most of the current contents of the MetStore.  Pieces of test
 code have been left as comments in the supporting units (especially
 the detection interrupt routine).}

program  Run_Obsv;                {Parent program to run Observe}
    uses Genutil, Crt, Dos;
{Runs the program Observe under field conditions.  This parent program
 provides the error handling and ensures the observing code keeps
 running no matter what.  Messages are written out to the log file
 C:\ Obsv.Log.  The program terminates normally on getting a Halt(999)
 from the child process.  Any values greater than this also terminate
 Run_Obsv.}
{$M $4000,0,0}      {16K stack, no heap required or reserved.}

program LookRate;                 {Observed rates in NZST_*. files}
    uses  OrbView, OrbDefns, Crt, Dos;
{Prints out the hourly rates at 20 minute intervals for the selected
 day's raw observation data.}

program CountRat;                 {Count meteor echo rates}
    uses    OrbReduc, OrbView, OrbDefns, GenUtil, Crt;
{The program runs through an observation file and provides a basic
 count of the meteor echo rates.  It uses the Home site profiles to
 calculate the maximum amplitude of the meteor echo.  It also records
 the range bin distribution.  This program looks at all the data
 detected at the Home site.  Originally the program also sorted output
 observations in an automated mode.  I found this was not entirely
 satisfactory and so introduced some more manual screening into the
 selection process.  The old versions are stored as CountRat.Old and
 CountRat.VII.  The rates information is output into the data
```

```
 directory as NZST_<hr>.Rat.  Some hardware error or bug in the
 observing program caused it to save two copies of observations on
 891121\NZST_08.  I have left a check in this program to ensure that
 all the observations in an observation data file are sequential.}

program ReDate;                    {Redate the data 6 May, 1989}
    uses  OrbRadar, OrbDefns, GenUtil;
{The hardware clock was set to the wrong date for the 1989 May 6 run.}
```

A.2.2 Historical Notes

```
program ObsvRun;                   {Execute a Meteor Observation Run}
    uses Dos,Crt,Graph,GenUtil,MetGraph,MetLift,MetIntrs,MetRadar;
{21 April 1989.  Starts up the radar, detects, lifts and saves to the
                 extended memory.  It then reads the observation
                 record and displays it on the screen.}

program ObsNew;                    {Get one meteor into AtoDMem}
    uses MetLift,MetGraph,MetRadar,MetIntrs,MetDefns,GenUtil,Crt,Dos;
{1989 May 1  0147hrs    Putting together the first observing routine}
{     May 6  0118hrs    A test routine diagnose the obsv records
                        with no apparent meteors in them.}
{     Sept 26  0942hrs  Rewriting so that it can run the detection
                        routine continuously.  The whole shebang.}
{This is included as an historical note.  Old code modules and
 assembling the program to run during the middle of an observing run.}

program TObsvOrb;                {Test observe orbits}
    uses  OrbGraph, Graph, OrbLift, OrbRadar, OrbDefns,
          GenUtil, Crt, Dos;
{1989 October 16 : Starting the program to test the new version of the
 detection routine.  One major change is the need to add the date onto
 the observation records when the data is written out to disk.
      October 20 : I have rearranged and transferred code into what I
 hope will be the final arrangement.  This new version of this test
 program is to see if it all still works as in TObsv22.Pas}

PROGRAM ECHOMAKE3(INPUT,OUTPUT,ECHOPARAM:FILE<KIND=DISK,MAXRECSIZE=120,
               BLOCKSIZE=1200,BLOCKSTRUCTURE=FIXED,SECURITYTYPE=CLASSA,
               SECURITYUSE=IO,NEWFILE=TRUE>,ECHOES:FILE<KIND=DISK,
               MAXRECSIZE=80,BLOCKSTRUCTURE=FIXED,BLOCKSIZE=2400,
               SECURITYTYPE=CLASSA,SECURITYUSE=IO,NEWFILE=TRUE>);
{Andrew Taylor          7 April,  1987}
{This program generates echo lookalike data that would come from the
 micro processor at Birdlings Flat.  The data is to be varied about
 the chosen values to simulate noise and get a handle on the
 uncertainties involved in the analysis.             }
```

A.3 Observing Code Test Routines

These programs test various aspects of the observing code. They all use the 400 radar sweep TestMax AtoDMem space that allows them to be run from within the interactive Turbo environment. They provide a flexible software base to diagnose any faults and for any future developments of the observing system.

```
program SetAtoD;                    {Test the SetAtoDMemory routine}
    uses  OrbRadar, OrbDefns, GenUtil, Dos, Crt;
{The program tests the SetAtoDMemory procedure located in the unit
 OrbRadar.  It prints out the address locations of the AtoDAlign
 buffer and then the first few sweep blocks of data.  The actual
 contents of the sweep blocks are then viewed using the procedure
 LookAtSweepBlocks.
 I had problems with the way the alignment buffer was set up when the
 computer came back from the importers with its new mother board
 (June 1989).  This program helped sort the problem out.  The heap
 manager seemed to be allocating new pointers on integral boundaries
 of 8 bytes instead of 16 as it should.  The (27 October 1989) of
 SetAtoDMemory deals with either situation.
 10 December 1989 : I have changed the way the alignment buffer is
 created.  It now uses the heap  pointer, HeapPtr to define the start
 of the AtoDMem.  On an absolute memory address that is a multiple of
 256 bytes.  This replaces the old system of allocating dummy buffers
 using the heap manager.  The new method is a simpler and more direct
 solution.}

program ExtMove;                    {Test moving to/from extended memory}
    uses  OrbLift, OrbRadar, OrbDefns, Dos, Crt;
{Sets up the extended memory addresses then tests the ExtMemMove
 routine.  The observation buffer ObsvBuff is successively filled
 with the storage number and then transferred up to the MetStore.
 These are then read down and the contents of the record are checked
 against the location numbers.  If an error occurs a test pattern is
 dumped out to the screen.  It prints out the value of the byte mod 10
 stored in each byte of the buffer; error locations are highlighted.}

program IntCount;                   {Counts IRQ#3 interrupts}
    uses  Dos, Crt;
{The program tests using a simple Turbo Pascal interrupt routine to
 service the interrupts sent from the timer control board.  Sets up
 the DMA interface card and the A/D timer control board.  It then
 uses an interrupt routine to count the number of IRQ#3's that come
 into the computer.  This can be compared with the actual number
 measured by a counter.}

program DMATest;                    {Test DMA acquisition of data.}
    uses  OrbRadar, OrbDefns, Dos, Crt;
```

```
{The program tests the use of the DMA controller to get the receiver
 amplitude data into the computer.  It sets up the AtoDMem space then
 runs an interrupt routine that services the DMA controller and counts
 the number of interrupts.  Once the run is stopped the contents of
 the AtoDMem can then be inspected with LookAtSweepBlocks.  One of the
 output lines on Port[DMACard] is taken high to allow timing of the
 service routine.}

program GetDump;                    {Get and save to disk an AtoDMem Dump.}
    uses  OrbTests, OrbRadar, OrbDefns, Dos, Crt;
{The program starts up the radar system, collects a full Test memory
 of sweep data and then writes it out to disk.  The test memory is
 small enough to allow running the program while in the interactive
 Turbo environment.}

program ShowDump;
    uses  OrbGraph, OrbTests, OrbRadar, OrbDefns,
          Graph, PostScript, Dos, Crt;
{Show the data in a TestMax sweep AtoDMem dump saved on disk.  The
 program first gives a movie of the video traces that would have been
 seen on the oscilloscope trace as the data was collected (slow
 motion).  This amounts to a view of the amplitude plotted against
 range for each sweep.  Second is an interactive longitudinal viewing
 of the data; amplitude versus time at a given range bin.  Third the
 program gives an interactive viewing of the phase data.  Finally the
 actual numerical data can be viewed.  The raw data files are assumed
 to come from C:\DATA\*.DMP }

program GetOneIn;                   {Get AtoDMem dump with a meteor echo.}
    uses  OrbGraph, OrbTests, OrbRadar, OrbDefns,
          GenUtil, Graph, PostScript, Dos, Crt;
{The program runs the radar system until a meteor echo is detected on
 the Home site.  It then collects another 300 sweeps of data and
 writes out the contents of the AtoDMem space to disk.  The data is
 viewed with AmpTrace to check that a meteor echo is present.  The
 AmplitudeDumps routine provides a different view of the data.  The
 program will run while in the interactive Turbo environment.
 Currently TestMax = 400, (31 Oct 1989) could be extended to 500 (?).}

program GetTime;                    {Gets time from CMOS clock}
    uses GenUtil, Crt, Dos;
{Tests the routines which read the time from the CMOS clock.  Reads
 and displays the time at 20ms intervals then reads off the date.
 Note the detection code uses the same interrupt service $1A.}

program TestDetect;                 {Test the interrupt Detector routine.}
    uses  OrbGraph, OrbLift, OrbTests, OrbRadar, OrbDefns,
          GenUtil, Graph, PostScript, Dos, Crt;
{The program loads an AtoDMem dump from disk and then runs the
```

```
 Detector routine over the data.  The detector routine is called by a
 software interrupt to inspect each successive sweep.  If the meteor
 profile occurs within the first 50 sweeps of the dump the quiet time
 operation of the detector will not be seen.  The collection program
 GetOneIn.Pas should ensure that this does not happen.  The program
 has several display sections
     (1)  The normal quiet time operation of the detector.
     (2)  Interactively, steps through sweeps where a detection occurs.
     (3)  Steps through where active screen is taken off and the
              detection is released to the handling program.
     (4)  Displays information stored by the detector at detection.
     (5)  Lifts observation based on the information in DetectInfo.
     (6)  Displays the complete Observation record.
     (7)  Views the AtoDMem contents.  Can check any DMAHold flags
              that should be set.}

program LiftTest;                {Test the LiftMeteorObsv routine.}
    uses  OrbGraph, OrbLift, OrbTests, OrbRadar, OrbDefns,
          GenUtil, Graph, PostScript, Dos, Crt;
{The program loads an AtoDMem dump loaded from disk and then runs the
 TestDetector routine over the data.  The LiftMeteorObsv routine is
 called.  The observation record obtained is compared with the data
 in the AtoDMem.  The program shows the following checks (5-7
 require a positive detection)
     (1)  A slow motion movie of the AtoDMem data is shown.
     (2)  Runs TestDetector then checks the contents of DetectInfo.
     (3)  If a meteor is released by the detector it is lifted out of
              the AtoDMem.  It must be detected before sweep 200.
     (4)  The contents of the SiteProfile can be dumped out.  This
              is commented out of the program.  The contents can be
              inspected using the Turbo debugger watch facility.
     (5)  Compares echo amplitude profiles lifted with the raw data.
     (6)  Presents phase data profiles compared with Home site data.
     (7)  Graphical display of the observation record recorded.
     (8)  A look at the contents of the sweep blocks.  Commented out.}

program ObsvTest;                {Observation Test Run.}
    uses  OrbGraph, OrbLift, OrbRadar, OrbDefns,
          GenUtil, Graph, PostScript, Crt, Dos;
{The program sets up the computer to run the orbit radar.  It runs
 the radar until one meteor observation has been detected, lifted and
 saved to MetStore.  The observation is then read down and displayed
 on the screen.  This provides a check that most of the observing
 routines are operating.}

program TReset;                  {Test the reset routine}
    uses  OrbRadar, OrbDefns, Crt;
{Does some minimal tests on the reset routine ResetVars in OrbRadar.}
```

```
program DiskFree;                {Checks space free on C: }
    uses  OrbLift, OrbRadar, OrbDefns;
{Calls the routine CheckDiskFree which checks that there is at least
 300 kBytes free on the hard drive C: for data.  Can't for the life
 of me work out why I set up the AtoD Memory space as well !? }
```

A.4 Hardware Testing

A.4.1 Receiver Characteristics

These programs were written to ensure a digital record of the calibration and bandpass information. I had the computer in the screened room to record the data. This was a summary exercise after my last maintenance/tuning effort with the 26.2MHz receivers; July/August 1989.

```
program CalRX;                   {Calibrate Receivers}
    uses  OrbRadar, OrbDefns, GenUtil, Dos, Crt;
{Runs a Turbo Pascal interrupt routine to load an A/D video sample
 into the computer memory.   This is used to calibrate the A/D video
 output scale with the radio frequency signal strength.  Running the
 whole system through the FM links for the remote sites and into the
 computer provides an overall check.  The data is written out to
 Temp.Dat and the original run copied as CalRX_89.Jul.  It can be
 graphed using CalDraw.Pas.}

program CalDraw;                 {Draw calibration curves for RX's}
    uses   OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{To graphically display the Calibration data stored in CalRX_89.Jul
 or CalRX_89.Aug.  The data was recorded using CalRX.Pas as the A/D
 video output for 1 uV steps up in rf input.  31 July  1989.  It can
 be used for future calibration tests using the same input format.}

program BandRX;                  {Measure Bandpass of the Receivers}
    uses  OrbRadar, OrbDefns, GenUtil, Dos, Crt;
{Runs the data collection hardware to measure the bandpass of the
 receivers.  This uses the signal generator in the screened room
 and steps through the 26.2MHz RX frequency range in 1kHz steps.
 The video output is digitised by the system and then recorded by
 the computer.  The data is output as Temp.Dat and the August 1989
 band pass measurement stored as BndRX_89.Aug.  It can be graphed
 using BandDraw.}

program BandDraw;                {Draw band pass curves for the RX's}
    uses  GenUtil, Graph, PostScript, Crt;
{To graphically display the Band pass data stored in BndRX_89.Aug.
 This program displays the six 26.2MHz band pass responses of the
 receivers.  It could be used for any future measurements of the
 radar receiver's frequency response.  The data was collected
 using BandRX.}
```

A.4.2 Aerial Power Distributions

These programs give plots of both the measured and theoretical power distributions for the main receiver and transmitter arrays.

```
program Show_Jup;                {Display Obsv_Jupiter information}
program Aerials;                 {Power profile Baileys Road.}
    uses   OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{To graphically display the power diagrams measured on Baileys Rd.
 The first in this series was collected as part of Roy Seaton's
 stage III project.  The latter one once the power feeds had been
 redesigned.  The data displayed comes from 1989 July 8, 14, 18,
 19 and August 31.  The last one is measured power distribution of
 the final aerial configuration.}

program RXPower;                 {Receiver array power distributions}
    uses GenUtil, Graph, PostScript, Crt;
{Draws azimuth and elevation power distribution diagrams for the
 Home site collinear receiver arrays.  For the remote receiver
 arrays the 16 wavelength arrays size needs to be reduced to 12.}

program TXPower;                 {Transmitter array power diagram}
    uses GenUtil, Graph, PostScript, Crt;
{The azimuth and elevation cross sections of the power distribution
 from the rhombic transmitter array.}
```

A.4.3 Celestial Radio Sources

Using the receiver aerials to detect celestial radio sources is a possible way of measuring the antenna power diagrams. Essentially this would give the beam width and direction of the receiver arrays. Unfortunately we could pick nothing out of the noise.

```
program Obsv_Sun;                {Observe the sun through the RX beam}
    uses  OrbRadar, OrbDefns, GenUtil, Graph, Crt, Dos;
{1989 Sept 29 : Putting together a program that will observe the
 passage of the sun through the RX beam.  The program will measure the
 incoming 26.2MHz radio noise from the sun, ie no TX and read an
 average noise value every second as the sun crosses the meridian.
 Despite the best of efforts this adventure did not work.}

program Show_Sun;                {Display Obsv_Sun information}
    uses  GenUtil, Graph, PostScript, Crt, Dos;
{Reads in the data file from a solar observing run and provides
 a graphics display of the results.  Some noise suppression and
 filtering will be necessary.  The data is stored in the Pascal
 record file Sol_1108.Dat.}

program Show_Jup;                {Display Obsv_Jupiter information}
```

```
    uses  GenUtil, Graph, PostScript, Crt, Dos;
{Reads in the data file from a solar observing run and provides
 a graphics display of the results.  Some noise suppression and
 filtering will be necessary.  Data stored as Jup_1121.Dat as a
 Pascal record.  1989 November 13.}
```

A.5 Orbit Data Reduction

A.5.1 Selection of NZST_<hr>.Orb files

The raw observation data stored on the PC needs to be prescreened. This semi-manual process gets rid of wind blown trails, hopelessly confused profiles and a number of sporadic-E echoes. The program Screen is run on the VAX to provide a preliminary sort. SortOrbs is an interactive graphics program which requires the user to positively identify observation records that are useful for reduction to orbital elements. This data selection process takes NZST_<hr>. files and produces NZST_<hr>.Orb files. Both sets of files are backed up by day directory on tape in the Physics department. The NZST_<hr>.Orb files for each month are also backed up at the Computer Services Centre.

```
program ScrnObsv(input,output,OKFile);
        {Reduce observations and get orbital elements}
{REWORKING TO RUN ON VAX MAINFRAME UNDER VAX PASCAL. 30 MAY 1990.}
{PROGRAM RECONFIGURED. 23 OCTOBER 1990.}
%INCLUDE  'Turbo.inc'           %INCLUDE  'GenUtil.inc'
%INCLUDE  'OrbDefns.inc'        %INCLUDE  'OrbReduc.inc'
%INCLUDE  'OrbLags.inc'         %INCLUDE  'OrbFres.inc'
%INCLUDE  'OrbElems.inc'
{The program is a modified version of CalcOrbs that does a
 preliminary screening on the raw observation files before a manual
 check to produce the *.Orb files used to reduce meteor orbits.  The
 program can be run from a command file Screen.Com held in the
 Operations directory [PHYS065.Operations].}

program SortOrbs;               {Sort observations into *.Orb files.}
    uses   NonPost, OrbDefns, Graph, Crt;
{The program looks through a data file and allows the user to draft
 the observations displayed into files for further processing.  The
 destination files are of two kinds, NZST_*.Orb and NZST_*.Sat.  The
 *.Orb file can be handled directly by the cross correlation routine.
 *.Sat files require timelags estimated using the rising edges.}
{31 Jan 1990.  I have decided that where the echo profiles are large
 enough the time lags between the stations will be calculated using
 the rising edge section only.  If any site has a maximum under 40
 then the full profile will be used for the cross correlation.  I only
 need to draft suitable observations into the one file NZST_**.Orb for
 the program CalcOrbs.Pas.}
{8 May 1990.  Removing the use of the PostScript unit from this
```

```
 program.  This involves fiddling with the units that the program
 uses.  A lot of other programs use them so it will have to be a
 temporary adjustment.  21 Nov 1990, changing to put copies of all the
 necessary routines into the unit NonPost.Pas so the program can be
 directly compiled and more easily maintained. }
{Adding in an option to save the data to a fresnel file for looking
 at atmospheric decelerations in more detail.  Want good fresnel
 oscillations on all three channels (ideally).  Pressing 'F' in
 response writes the observation to both the *.Orb and *.Frs files.}
{Adding a file to contain 'high precision' observation profiles.  Ones
 with a clear and well defined rising edge on all three channels.}
{Taking out the high precision option.  Does not seem to be necessary.
 I will work on automating a program to select good observations.  I
 am retaining the fresnel file at the moment and a special file to
 put interesting observations into.}
{21 November 1990:  Removing the option to put observations into the
 Fresnel file.  It is better to pick these things out using the
 automated search on reduced data.  I am adding in the max rise slope
 routine as a pre-screening step in the reduction process.  This will
 output as NZST_<hr>.Scn so SortOrbs needs to sort *.scn files.}

MODULE Checkout(INPUT,OUTPUT);
{Andrew  Taylor  11 August   1987
     mod. 17 Feb  1988,  26 July 1988  To Checkout alone.
Routine to do initial checking on raw echoes from Birdlings Flat.
1990 December : This file holds routines that do clever things with
the 1988 eta aquarid data run.  This was still using the tape drive
and things ? You know the EPROMS.  It is written in VAX/VMS Pascal
in the days before the PC.}
```

A.5.2 Reduction to Orbit_<hr>. files

A version of the program CalcOrbs is held on the VAX. This does the bulk of the data reduction calculating Orbit_<hr>.Txt ReducedData record files. These files along with the Report.Dat files are backed up on tape at the Computer Services Centre. The *.Txt files are transferred to the AMOR386 PC and converted to Orbit_<hr>. files by Pas_Txt. A copy of these Orbit_<hr>. files is held on floppy. The value used for the phase calibration constant by CalcOrbs needs to be checked with PhaseCal. The data can be viewed using DistDay.

```
program CalcOrbs;          {Reduce observations to orbital elements}
    uses  OrbElems, OrbFres, OrbLags, OrbReduc, OrbDefns,
          GenUtil, Crt, Dos;
{The program reads in data from <name>.Orb files.  It then reduces
 the observation records and calculates the orbital elements of the
 original meteoroid.  Information is put out to Orbit_<hr>.ReD as a
 ReducedData record.
14 May 1990 : The program has been modified to search for all the
```

```
 *.Orb data files within the named directory and process the lot.
 Output to Orbit_<hr>.
13 October 1990 : Modified to use the new max rising slope method for
 determining the timelags between the stations.  ReducSweep has been
 defined as a constant in OrbReduc; should reject a few less now.  New
 deceleration formula for the atmospheric correction is introduced.
 Determining the decay constant now uses the differentiated profile.
 Also using the despiked observation profiles for the Fresnel
 calculations.  This is a definite 11th hour mod.  The program also
 contains a drafting option to check its decisions about what to
 include.  Where Fresnel velocities are calculated on more than one
 channel an additional check is made to ensure that they agree to
 within 5km.s-1 before an atmospheric deceleration is estimated.}

label   999,    {Error condition.  Cannot get orbits.}
        888,    {Routines could not get good aeronomy data.}
        777;    {User terminates the program with  S  or  Q .}

program CalcOrbs(input,output,Report,Outdata);
        {Reduce observations and get orbital elements}
{REWORKING TO RUN ON VAX MAINFRAME UNDER VAX PASCAL. 30 MAY 1990.}
{PROGRAM RECONFIGURED. 23 OCTOBER 1990.}
%INCLUDE  'Turbo.inc'           %INCLUDE  'GenUtil.inc'
%INCLUDE  'OrbDefns.inc'        %INCLUDE  'OrbReduc.inc'
%INCLUDE  'OrbLags.inc'         %INCLUDE  'OrbFres.inc'
%INCLUDE  'OrbElems.inc'

program Pas_Txt;                {Writes Orbit_**. from Orbit_**.Txt }
    uses  {OrbView,} OrbReduc, OrbDefns, Crt, Dos;
{Takes the reduced data text output files from the vax and writes
 Turbo Pascal output files of ReducedData record type to use on the
 PC.  An executable image with menu selection is held in C:\Useful.}

program PhaseCal;               {Check PhaseCalib const is O.K.}
    uses  OrbView, OrbDists, OrbReduc, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{The program displays the geometric altitude distribution to check
 that the PhaseCalib constant used by CalcOrbs to produce the
 Orbit_**. file was reasonable.  There is not enough room in memory
 to hold the Turbo environment and compile this program with the
 OrbView directory and file selection menus.  The StartHour and
 EndHour strings allow the day to be split into sections.  The
 combination of both the Home site geometric altitude plot and the
 phase angle plot should ensure against misconnected aerials as well.
 The variable DeltaCalib can be used within the program to make fine
 adjustments to the assumed PhaseCalib value.  More negative values
 push the mean altitude up.  More positive lower the mean altitude.
 Mostly I aim for a peak at around 97 km.  There seems to be a fairly
 sharp cutoff at about 100,102 km and most of the meteors are detected
```

```
 between 90 and 100 km.  5,10,25,50,75,90,95% levels are marked on the
 plot along with the average.  A compiled copy is held in C:\Useful.}

program Re_Calc;              {Recalculate reduced data records.}
    uses  OrbElems, OrbDefns, GenUtil, Crt, Dos;
{The program provides a framework from which the Orbit_<hr> files
 from one day can be reprocessed.  The new data is output as *.new.
 The program will usually process the default directory.
13 November 1990 : I have corrected the longitude of perihelion
 variable.  It should be in radians but I incorrectly used a degrees
 longitude of the ascending node to evaluate it.  It should be based
 on 1950.0 LongAsc1950 radians measure.}

program Check_Ranges;         {Check and eliminate zero ranges.}
    uses  OrbDefns, GenUtil, Crt, Dos;
{Have found several reduced data buffers with remote site range
 entries of zero.  These are searched for and eliminated from the hour
 record.  I do not understand why or how these occur. This is probably
 a check that should be included in the ScrnObsv.Pas programs.}
```

A.6 Data Reduction Test Code

A.6.1 Range Determinations

This code section is a very recent addition (1991 May 12). It uses the amplitude of the adjacent rangebins to determine an offset of the peak echo location and hence give more accurate range. It could be that this program belongs in the Observing Code Test Routines section, Section A.3.

```
program RangScan;             {Fit to get range scan offset.}
    uses   OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{Begun 1991 April 23.  I should not be playing around at this point !
 The program uses the output shape from the Home site receiver of
 a 66uS pulse to model as a measure of the reflected radar pulse.  The
 location of the 40uS range bin sampling points is shifted relative to
 this.  The program has first been used to give the least squares sums
 to do the peak amplitude fit and the range scan offset.  The program
 will then test the range offset calculation scheme.  The scheme has
 been set to use the quarter, half, quarter amplitude of that sweep as
 the calibration.  The relation peak amplitude = 1.67*Recorded
 amplitude is assumed.  Once the range scan shows any evidence of peak
 saturation the offset fit should be abandoned.  The new offset
 information will be stored in the DetectSite byte of the ObsvBuff
 replacing that use for it.}
```

A.6.2 Amplitude Profiles

The echo amplitude recorded by the observing program is not the peak amplitude for the echo pulse. The first two programs show the relationship between

the two and give a calibration curve for it. Some signal processing on the amplitude profiles is required to allow the reduction program to work on raw observation data. The smoothed profile is used to determine the maximum amplitude. The rise and decay times of the echo profile both use the differentiated profile. DecayDif explicitly tests the decay finding routine.

```
program Amp_Cal;                {Calibration recorded amplitudes}
    uses   OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{Produces a calibration curve for the 1/4, 1/2, 1/4 averaging that is
 implicit in the recorded data profiles.  This curve plots recorded
 amplitude against peak amplitude of the echo pulse.  The recorded
 amplitude becomes non linear at about 160 A/D units.}

program Amplitud;               {Look at 1/4, 1/2, 1/4 averaging}
    uses   OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{Begun 1991 April 19.  I should not be playing around at this point !
 The program uses the output shape from the Home site receiver of a
 66uS pulse to model the reflected radar pulse.  The location of the
 40uS range bin sampling points is shifted relative to this.  The
 amplitude resulting from the 1/4, 1/2, 1/4 average that is the
 recorded as the echo amplitude is calculated for these relative
 offsets.}

program TDeSpike;               {Test the DeSpikeProfile routine}
    uses   OrbReduc, OrbGraph, OrbDefns, Graph, PostScript, Crt;
{1990 May 2 has some nice corona in the amplitude profile that the
 despike routine can work on (NZST_02.Orb).}

program TSmooth;           {Test the smoothing routine; TriSmooth}
    uses   OrbReduc, OrbGraph, OrbDefns, Graph, PostScript, Crt;

program DecayDif;               {Decay constants using differential.}
    uses  OrbLags, OrbReduc, OrbDefns,
          GenUtil, Graph, PostScript, Crt;
{By looking at the differential of the amplitude profile it should be
 possible to pick a better start point for the decay constant.  It may
 be possible to detect wind affected profiles and reject the decay
 calculation on them.  This program tests the GetDecay routine and
 provides a diagnostics display of the output.}
```

A.6.3 Time Lags

The time lags are determined by comparing the point of maximum rising slope on the rising edge of the echo profiles from the three sites. This final method was adopted as being least affected by atmospheric influences. Considerable effort had been expended on producing cross correlation routines to calculate the timelags. These tended to get rather ad hoc for long lasting persistent trails and heavily dependent on the positioning of the window used to select

the rising edge section in strong 'pre-Fresnel' profiles. A number of other timelag determining options were also tested including picking points on the rising edge and running the detection algorithm with various detection levels. These can all be compared using the program LagsList.

```
program MaxSlope;                {find Maximum Slope of rising edge.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{This program is a test bed to develop a routine to look at the
 smoothed profile and find the point at which the rising edge reaches
 its maximum slope.  This is calculated in the routine MaxRiseSlope
 from the OrbLags.Pas unit.  The approach is a possible alternative to
 the XCorr time lags method.  The display routine also places the rise
 time information onto the plot.}

program MaxLags;                 {Time lags from maximum slope.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{Program tests the maxslope of rising edge timelag routine.  The
 routine is called MaxRiseLags and found in OrbLags.Pas.  The
 program processes the max rise slope time lags in the named file
 until it detects an error.  It then drops into a diagnostics display
 routine to present the information.  It is possible to stop the
 routine by pressing Q at any time.  Test data is held in the file
 C:\DATA\ TestLags.Dat.}

program LagsList;                {Compare different timelags methods}
    uses   OrbLags, OrbReduc, OrbGraph, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{The program reads in observation data and carries out a series of
 time lag determinations to check the performance of the cross
 correlation routine.  The results can be directed to the file
 Lags.Dat to act as a data base for comparing different methods.
 Currently they go directly to the screen, FileName:= ''.  The
 following methods are compared :
     Detection points, LappLag.
     Detection points at 30 A/D units.
     Max points off the TriSmoothed curve.
     Rise Point lags, average of 50,60,70,80% points.
     Maximum rise slope comparisons.
     XCorr with full profiles.
     XCorr with rising edge only profiles.
21 Nov 1990: The timelag determining scheme finally chosen was the
 max rising slope method.}

program TXCorr;                  {Test the XCorr and Shift routines}
    uses   OrbLags, OrbReduc, OrbGraph, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{The program provides a test bed to run the XCorr routine on
 manufactured profiles.  It currently provides two square profiles
```

```
 10 pulses apart.  The cross correlation routine is used to calculate
 the timelags between them.  Two correlations are done; the first
 directly on the profiles and the second with zero shifted to the
 mean amplitude level.}

program LagShift;                {Test the XCorr with window function}
    uses   {OrbView,} OrbLags, OrbReduc, OrbGraph, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{The program reads in observation data and produces a series of
 graphical displays to check the performance of the cross correlation
 routine.  It runs the routine to find the lags between the Home and
 Nutt sites.  This would be relatively easy to modify.  A cross
 correlation is done with both the full profile and the rising edge
 section only.  XCorr is no longer part of the data reduction scheme.}

program RisePnts;                {test RisePoints routine.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{This program is a test bed to develop a routine to look at the
 smoothed profile and find the 50%, 60%, 70% and 80% points of maximum
 amplitude.  These can then be compared to give an estimate of the
 timelags between the three stations.}

program RiseLags;                {Timelags from rising edge points.}
    uses   OrbLags, OrbReduc, OrbGraph, OrbDefns,
           GenUtil, Graph, PostScript, Crt;

{The program reads in observation data and tests the RisePointLags
 routine output.  The estimated lags are output to the screen.  This
 can be redirected to say 'Lags.Dat'.  The lags are based on an average
 of the 50,60,70,80% points of the maximum amplitude.}
```

A.6.4 Phase Angle

These programs test/display aspects of determining the elevation angle of the radar echo from the meteor trail. Most of the effort is involved in determining the phase angle and resolving any phase ambiguities due to the Home site aerial separation. The susceptibility of the phase calibration constant to weather is a continuing source of problems. Some long term change is expected as the coax transmission lines degrade.

```
program Plot_Cal;                {Wet wire phase calibration values}
    uses  OrbDists, OrbReduc, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Provides a diagram showing the dependence of the phase calibration
 constant on the weather conditions.  Eventually (some time) we will
 add a dogleg to hopefully eliminate the dependence on the wet wires.}

program CalConst;                {Get Phase Calibration constant}
```

```
    uses  OrbReduc, OrbDefns;
{A development program to read in a phase constant array for the
 calibration constant to be used for that day.  This routine will
 require the PhaseCal.Dat data file to be updated observations
 that are processed on each new day.}

program FitPhase;                {Display empirical FitPhase_1983 fit.}
    uses  OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{To graphically display the empirical FitPhase_1983 routine.  The
 field data from Nov 1983 is also plotted.}

program TGetPhas;                {Test the Get_Phase routine}
    uses  OrbPost, OrbReduc, OrbDefns,
          GenUtil, Graph, PostScript, Crt;
{The program produces a graphical display as a check that the routines
 associated with calculating the phase angle give sensible results.
 It shows the points used to give the average Tin and Tos values.  The
 phase from each radar pulse is also calculated and graphed for
 comparison with the smoothed Home site amplitude profile.  Finally
 the calculated phase value is marked on the phase profile box.  A set
 of test data is located in the file C:\DATA\TGetPhas.PEN.  The
 program can easily be modified to run on any observation file.}

program TResolve;                {Test ResolvePhase in OrbReduc.Pas}
    uses OrbLags, OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{The routine ResolveAmbiguities (OrbReduc.Pas) makes no use of the
 time lag information in resolving the phase ambiguities.  This is
 necessary where only the Home site profile is available as in the
 aeronomy quality observations.  Where the time lags have been
 calculated a more definitive resolution of the phase ambiguities can
 be made.  This program is being started as a test bed for the routine
 ResolvePhase.  The routine uses Lag23 to decide whether the meteor
 comes from the north (0 < Elev < 90) or from the south
 (90 < Elev < 180), ie echo comes from behind.  The program uses data
 that has come from the manually drafted NZST_<hr>.Orb files.
 14 April, 1990 : The ResolvePhase routine has been rewritten to deal
 more completely with ambiguities in elevation and range.}

program ElevCurv;                {Phase Range relation given elev}
    uses  OrbReduc, OrbDefns, GenUtil, Graph, Crt;
{To provide a data file of phase angle, elevation and range for
 various constant altitude curves.  These use the curved Earth
 formula.   This data is used to plot the 100km altitude curves on
 the phase range diagram.  The data set is stored in ElevCurv.Dat.}

program PhasRang;                {Phase, Range distribution}
    uses OrbLags, OrbReduc, OrbDefns,
         GenUtil, Graph, PostScript, Crt, Dos;
{The program draws the phase range distribution for the reduced data
```

```
 from one day's run.  It can be broken down into subsets with StartHour
 and EndHour.  This program plots the phase and range directly from
 the Orbit_**. files.  Therefore those observations with ranges
 greater than 400 km, an ambiguous range or altitudes too high
 (Alt > 120 km) or low (Alt < 70 km) will not appear on the plot.
 More or less you will see what you have selected to see.  The program
 RawPhasR.Pas uses a raw observation file and so allows the data to be
 viewed without the various selection effects that processing imposes.
 This program puts on a set of overlays showing the range phase
 relationship for meteors at a particular altitude (100km).}

program RawPhasR;                {Raw NZST_**. data Phase Range diagram}
    uses  OrbView, OrbReduc, OrbGraph, OrbDefns,
          GenUtil, Graph, PostScript, Crt;
{Takes a file and produces a range phase plot for the data in an hour
 file.  It puts on a set of overlays showing the range phase
 relationship for meteors at a particular altitude (100km).  It just
 plots the range bin instead of the range for the data points.
 23 June 1989 : Playing with the program to get nice new theoretical
     curves.  Aerial spacing of 4.96 wavelengths and making 0 phase
     difference as an even number of 2 pi down from overhead.
 22 May 1990 : Taking this program as a base to plot out the phase
     range diagram for one day's data run.  This I need to do to check
     that the phase constant is correct.
 19 October 1990 : The program is now a diagnostics routine and
     produces the phase range diagram for one raw observation file.
     The phase calibration constant is checked by PhaseCal.Pas using
     the altitude distribution.  This program cannot be compiled from
     within the Turbo environment.  A copy is held in C:\Useful }

program OldFitPhase;             {Display empirical OldFitPhase_1983}
    uses  OrbReduc, OrbDefns, GenUtil, Graph, PostScript, Crt;
{Display the empirical OldFitPhase_1983 routine.  The field data from
 Nov 1983 is plotted along with the OldFitPhase_1983 results.}

program PhasDist;                {Phase distribution from FitPhase.}
    uses  OrbDists, OrbReduc, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{The program displays the phase angle distribution without being
 inverted or having the phase calibration constant added.  This allows
 the distribution to be more easily compared with the phase fitting
 routine.  Data from 1990 June 4 (D:\900604) provides the distribution.
 Either the FitPhase_1983 or the OldFitPhase_1983 routines can be used.}
```

A.6.5 Fresnel Diffraction

A set of test bed programs for the velocity and deceleration routines using Fresnel diffraction methods. The final two programs test procedures that have been dropped from the Fresnel reductions.

```
program Fres_SMA;                {Fresnel test bed for SketchMeanAmp.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{This program is a test bed to try and find a better averaging
 function to use in conjunction with the Fresnel velocity
 determination routines.  The new routine does it fairly well.}

program Fres_SHC;                {Fresnel test bed for SizeHalfCycles.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;

program Fres_GIS;                {Fresnel test bed for GetInvSinY.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;

program Fres_FEC;                {Fresnel test bed for FindEndCycles.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;

program Fres_Atm;                {Fresnel Atmospheric decelerations.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{Tests the least squares routine that fits an atmospheric
 deceleration to Fresnel data for more than one station.  This
 calculation is being combined into an additional procedure
 GetFresnelVelocities.}

program Fresnel;                 {Fresnel velocity displays.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{Full three station Fresnel stuff.  The program works through a file
 of Observation records.  It calculates Fresnel velocities (if any
 exist) and then displays a series of velocities corresponding to sub
 domains of the oscillations.  This is a useful guide to estimating
 atmospheric decelerations.  There is some evidence that meteors are
 slowing as they travel through each successive Fresnel zone but
 quantifying the deceleration is not easy.  It is also fairly
 variable.  The program then shows the individual Fresnel patterns
 and velocity sub domains.  The individual decelerations can be
 compared with the total atmospheric deceleration.  I don't bother
 saving the individual values any more; too variable and often don't
 agree at all.}

program TestAv;                  {Test AverageVelocities,  OrbFres.Pas}
    uses OrbFres, Crt;
{Tests a fairly clever averaging routine.  It takes notice of the
 weight attached to each value and makes decisions about which values
 to ignore.  This program allows the values and weights to be adjusted
 to check the logic is making the expected choice.  The routine is
```

```
 used to check that the Fresnel velocities from various sites agree.}

program Fres_Bob;              {Output pre-Fresnels for Bob.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{Produces a set of postscript amplitude profile diagrams on a set of
 expanded scales.  These were to look at the pre-Fresnel
 'fragmentation' (?) effects.}

program Fres_CVS;              {Test bed for CheckVelocityStable.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{The CheckVelocityStable routine is not used any more.}

program Fres_GFD;              {Test the GetFresnelDecel routine.}
    uses   OrbFres, OrbLags, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{I do not now measure the deceleration on only one profile.  Too
 subject to the vagaries of the end point selection.  Results are
 not accurate enough.}
```

A.6.6 Orbital Elements

```
program TOrbit;                {Test the GetOrbit routine}
    uses   OrbElems, OrbReduc, OrbDefns, GenUtil, Crt;
{The timelags, elevation angle and time of observation can all be
 entered with complete freedom.  The program calls the GetOrbit
 routine in OrbElems and directs the diagnostic notes and results
 to the file Reduc.Dat in the default directory.}

program TMatMul;               {Test the Matrix Multiply}
    uses   OrbElems, OrbDefns, Crt;
{Tests the matrix multiplication routine used to calculate the
 heliocentric meteor velocity.  The program also checks out the
 arctan function.}
```

A.6.7 Record to ASCII conversions

```
program TPas_Txt;              {Test convert of pascal to text files.}
    uses  OrbReduc, OrbDefns, Crt, Dos;
{The program Pas_Txt takes the reduced data text output files from the
 vax and writes Turbo Pascal output files of ReducedData record type
 to use on the PC.  This program tests the WriteBuffer_Txt and
 ReadBuffer_Txt routines.}
```

A.7 Data Management

A.7.1 Shifting Data Around

These programs identify particular observations to be copied into a separate data file. I typically use this approach to reduce the amount of data to be waded through when looking at something interesting. The programs look at reduced data records to note either ReducedData or Observation records to be lifted. The particular records chosen are stored in C:\DATA*.Dat files along with any relevant notes. The actual data is held as *.Pen files in the same directory.

```
program AutoNote;                {Automatic Noting into Temp.dat file.}
    uses   OrbDefns, GenUtil, Crt, Dos;
{The program goes into the specified reduced data directory and picks
 out selected observations as of interest.   The date, time and run
 number is noted in the output file Temp.Dat.  The program puts the
 Temp.Dat file into a data directory C:\DATA.  I use this directory
 to store the information files *.dat output from this program and
 the observation files *.pen that are associated with them.  The
 required conditions need to be programed into this before it is run.}

program AutoGet;                 {Automatic Get Observation records.}
    uses   OrbDefns, Graph, Crt, Dos;
{The program uses the GetObservation procedure to find selected
 observation records and write them out to the file C:\DATA\Temp.Pen.
 The directory C:\DATA is used to store the information files *.dat
 input files and the observation files *.pen output by this program.}

program AutoRedu;                {Automatic Get ReducedData records.}
    uses   OrbDefns, Graph, Crt, Dos;
{The program uses the procedure to find reduced data records noted in
 Temp.Dat and write them out to the file C:\DATA\Temp.Pen.  The
 directory C:\DATA is used to store the information files *.dat and
 ReducedData files *.pen output by this program.  Notice that these
 *.Pen files are not observation records but ReducedData records.  The
 extension *.Pen is a carry over from using 'drafting' programs.}

program Draft;                   {Draft observations to Interest.Pen }
    uses   OrbView, OrbGraph, OrbDefns, Graph, Crt;
{The program runs through the selected observation file and allows
 the user to add observations into a new Interest.Pen file.  This will
 stay in the current default directory.  If Interest.Pen already
 exists then the observations records will be appended to the file.
 Extra drafting pens could be added.  A version can be compiled to run
 with a menu selection of the raw data file to be looked at.}
```

A.7.2 Observation Display Routines

```
program LookDisk;                {Look at observation data on disk}
    uses   OrbView, OrbDefns;
{The mask used to search for different file extensions is different on
 the two machines.  AMOR386 is NZST_<hr>.Orb.  AMOR286 is NZST_<hr>.}

program LookObsv;                {Look at an observation file}
    uses   OrbPost, OrbDefns, Graph, PostScript, Dos, Crt;
{The program provides a bit more freedom to enter file names and look
 at various files (in unexpected places) of observation records.  It
 is a lift from the routine DisplayDiskObsv.  I tend to use this more
 often on AMOR386.}

program PostObsv;                {PostScript run on observation file.}
    uses   OrbPost, OrbDefns, Graph, PostScript, Crt, Dos;
{Takes an observation file and plots each observation record on the
 screen and produces a postscript file that can then be printed on the
 NEC.  The input file needs to be entered.  At the moment I can't
 remember how to rename or copy the output file from within Turbo.
 Where is my reference manual ?  Lost, Stolen or just sulking.}

program Look_Orb;                {Looks at NZST_*.Orb data on disk}
    uses   OrbView, OrbDefns;
{Views the NZST_<hr>.Orb files on the AMOR286 luggable machine.}
```

A.7.3 Data Display Options

```
program ReducOrb;                {Reduce observations and display}
    uses  OrbElems, OrbFres, OrbLags, OrbReduc, OrbGraph, OrbDefns,
          GenUtil, Graph, PostScript, Crt, Dos;
{This program is a conglomeration of the CalcOrbs.Pas and LookDisk.Pas
 programs.  It is intended to provide a postscript output file which
 displays all the information about an observation on one page.  This
 includes a graphics display of the observation record and the
 contents of the reduced data record.  For a start I'll just make it
 do one observation at a time.}

program Frag;                    {Plot Harvey amplitude profile.}
    uses   OrbDefns, GenUtil, Graph, PostScript, Crt;
{Program to take the output file from Harvey's meteor reflection model
 program and give a graphical display of the amplitude profile.}
```

A.7.4 Menu File Selection

```
program NewDir;                  {New select directory routine}
    uses   OrbView, OrbDefns, Unit_Mnu, Unit_Key, Crt, Dos;

{Tests the new SelectDataDir routine from OrbView.  This one uses a
 file mask to search for specified files.  It beeps if there are no
```

```
files of that type in the selected directory.
```

A.8 Atmospheric Deceleration

The relationship between atmospheric deceleration and meteor velocity has been investigated. These programs use the information calculated from the Fresnel diffraction patterns. The data sample is therefore biased toward the lower velocity meteors detected by the radar. The meteor ablation coefficient can be estimated from the deceleration velocity relationship.

```
program DecelVel;                {Plot Deceleration vs Velocity}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the Orbit_**. ReducedData files and produces
 a graph of Deceleration against Fresnel velocity.  Note the Fresnel
 velocity will tend to give the low group of the time lag velocities.
 The program writes an output file Temp.Dat that contains the
 deceleration, velocity and zenith angle for observations where it
 has been calculated.  This data file can then be used in the other
 Decel***.Pas programs.}

program DecelFit;                {Theoretical fit of Decel vs Velocity}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the C:\DATA\ DecelVel.Dat file.  This high
 quality data is used to fit a power law relationship between the
 atmospheric deceleration of the meteor and its Fresnel velocity.
 The scatter plot only includes 6/7 of the data points.  Two power law
 relationships are plotted with all the variation included in one and
 then the other.  Mean point for 10km.s-1 and 10km.s-2 bins are
 plotted as a check on the regression curves.  I added a linear fit to
 the data as well.  It seems to do reasonably well.  To test the
 d = (v)^1.6 relation the program DecelLog.Pas is used.  The log-log
 plot lets you get away with more! }

program DecelLog;                {Theoretical log fit, Decel vs Vel}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the C:\DATA\ DecelVel.Dat file.  This high quality
 data is used to fit a power law relationship between the atmospheric
 deceleration of the meteor and its Fresnel velocity.  The scatter
 plot only includes 6/7 of the data points.  In fact most of the 1/7
 points not in the region are negative decelerations (most near zero)
 so could not be part of a log fit anyway.  The log-log scatter plot
 shows the theoretical power relation is at least plausible.}

program DecelSig;                {Direct fit of Ablation coefficient}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the C:\DATA\ DecelVel.Dat file.  This high quality
 data is used to estimate the ablation coefficient.  The relation
 assumes that the deceleration is at the maximum ionisation.  It does
```

```
a least squares fit to get the sigma, ablation coefficient.  The
actual zenith angle of the observation is used.}
```

A.9 D-Criteria Searches

I have implemented only the Drummond modified D-Criteria search routine. The Southworth-Hawking D-Criteria could also be used. In the end I picked up η Aquarid meteors by a direct inspection of the orbital elements of their associated orbits. These went into the C:\ DATA \ Allbox.eta file.

```
program DCrits;                 {Test the D-Criterion functions}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Test the D-Criterion function DDcrit (and potentially the SHcrit
 function) from the OrbAstro.Pas unit.  Plots on an R.A. Dec diagram
 those orbits within DD < 0.20 of the specified test orbit.  The
 longitude of the ascending node is set 'free'.  The distribution of
 DD values for the day are graphed as a final output.}

program DCritEta;               {eta Aquarid D-Criterion searches}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Looks through all the data files from 0300 to 0700hrs (Orbit_03 to
 Orbit_06) and compares the observed orbits with the prescribed stream
 orbit.  The number of meteors within a particular value of the
 D-criteria is recorded in the file DD.Dat.  I have renamed these as
 DDHal.Dat, DDeta.Dat and DDnew.Dat.  This program basically does a
 stream search after the scheme of Duncan Olsson-Steel.  ShowDD.Pas
 produces a graphical display for this data.}

program ShowDD;                 {Display the data from DD.Dat}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Takes the output from DCritEta and plots it in the ADT Longitude
 range.  This data is stored as DDeta.Dat, DDnew.Dat and DDHal.Dat.}

program DCriTest;               {Test the D-Criterion functions}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Test the D-Criterion function DDcrit (and potentially the SHcrit
 function) from the OrbAstro.Pas unit.  Uses just two test orbits.
 Written 1991 Feb 28 because Jacks southern delta Aquarids stream was
 not giving good answers.  Turns out the program DCritEta.Pas had the
 stream orbit long ascending node set to the solar longitude
 (descending node detection).  Should be set to the longitude of the
 ascending node for more general use.
        StreamOrbit.LongNode:= Buffer.LongAsc1950*DegToRad;
 For the eta Aquarids the stream occurs at the descending node so the
 solar longitude is just the longitude of the ascending node.}

program DDSerial;               {Serial stream search on eta Aquarid}
    uses  OrbDefns, GenUtil, NonPost, Graph, Crt, Dos;
{To conduct a serial search for eta Aquarid stream members.  The
```

```
membership requirement is :
      Every member must be with DD < 0.04 of two other member
or    Must be within DD < 0.04 of the mean stream orbit.
The program reads in the data set in C:\DATA\AllBox.eta does the
search and then writes out identified members to Stream.eta}
```

A.10 η Aquarid stream

```
program Sort_Eta;                {eta Aquarid from R.A. Dec box.}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Looks through the files from 900428 to 900518 and sorts out a set of
 eta Aquarid meteors with 320 < R.A. < 360 and -9 < Dec < 4.  These
 are stored into the C:\DATA directory as AllBox.eta.  This should
 speed up looking at the orbital distributions.  The program also
 plots a right ascension versus declination plot as it selects.}

program Eta_rad;                 {Plot R.A. Dec for Aquarid.eta}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the AllBox.new or AllBox.eta ReducedData files and
 produces a set of graphs showing Right Ascension versus Declination on
 an enlarged scale.  The program can also be used to lift out the
 Aquarid.eta file.  The file AllBox.new uses the recalculated orbits
 based on a lower atmospheric deceleration.}

program Eta_DD;                  {Plot R.A. Dec for Drummond eta Aquarids}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the AllBox.eta ReducedData file and produces a
 Right Ascension versus Declination diagram.  Those meteors which come
 within DD < 0.10 of the new ADT eta Aquarid orbit are marked on as
 larger spots.  The sloping R.A. declination box is plotted on the
 diagram.  The data comes from C:\DATA\AllBox.eta. }

program Eta_Reprocess;           {Reprocess eta Aquarid orbits.}
    uses  OrbElems, OrbDefns, GenUtil, Crt, Dos;
{The program reads in data from the AllBox.eta file.  It then
 reprocesses the reduced data records with a modified OrbElems unit.
 This program was modified from CalcOrbs.Pas, 8 Nov 1990, to reprocess
 data to try and match a better value of atmospheric deceleration to
 the eta Aquarid shower meteors.  I have changed it from 0.80*V^(1.60)
 to 0.27*V^(1.60).  It will reduce the velocities for the eta Aquarids
 by about 3km.s-1.  Notice that this will change the corrected right
 ascension and declination of the meteors slightly.  The relevant
 piece of OrbElems needs to be changed before running this program.
 The file C:\DATA\AllBox.eta is reprocessed to AllBox.new.  In fact
 all orbit observations now use the 0.27 coefficient.}

program DistEta;                 {Orbital Distributions for Aquarid.eta}
    uses  OrbAstro, OrbDists, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
```

```
{Reads in data from the AllBox.new ReducedData file.  It then produces
 the required cumulative distribution diagrams.  The interactive
 display section has an inbuilt help.  Double key strokes must be
 pressed within 2 seconds of each other to be useful.  The data being
 viewed has been preselected using SortEta.Pas.  The file Aquarid.eta
 contains the meteor orbits finally selected as belonging to the
 stream.  The program provides options to view different selection
 processes.  The fourth in the list is the final one used.}

program EtaDDRAD;                {Plot R.A. Dec for DD members}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the DD_Aquar.eta ReducedData files and produces
 a set of graphs showing Right Ascension versus Declination on an
 enlarged scale.  The program can also be used to lift out the two
 small sub-radiants that have appeared in the data.  1991 March 5.}

program Eta_Lot;                 {longitude for all obsv 900428-900518}
    uses  OrbAstro, OrbDists, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Draws the total number of observations made on each day during the
 1990 eta Aquarid shower onto a longitude plot.  The number of
 particles detected at their descending node is also plotted.  Note
 that this means the solar longitude equals the ascending node.}

program EtaLind;                 {Plot Lindblad eta Aquarids radiants}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Radiants from a review by Lindblad of all the eta Aquarid meteor
 radiants detected up to 1988.}

program EtaOrbit;                {Average orbital elements.}
    uses  OrbAstro, OrbDists, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{The orbit for the eta Aquarid meteor stream is poorly determined.
 This program estimates the average orbital parameters from the set
 of orbits contained in the file C:\DATA\ Aquarid.eta.  The data
 is written out to the file Show.Dat.}

program EtaShift;                {Plot radiant shift for Aquarid.eta}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the C:\DATA\ Aquarid.eta ReducedData files and
 produces a graph showing the daily shift in Right Ascension versus
 Declination on an enlarged scale.  It plots the daily average
 radiant position.}

program Eta_Vh_i;                {Inclination vs heliocentric velocity.}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the C:\DATA\AllBox.eta ReducedData file and produces
 a scatter plot of inclination versus heliocentric velocity for non
 shower meteors.  This is to see whether the excess in each of these
```

```
parameters is around the value of dispersed shower meteors.
1991 March 8.}
```

A.11 Orbit Density Cross Sections

A series of orbital density cross sections centred around the η Aquarid stream orbit were done. The first three programs do this taking care that the test orbits intersect with the Earth at all points across the cross section. This was done by varying components of the heliocentric velocity. The other two programs, Thes_*, were my early attempts to produce cross sections. The sections were done by varying each of the orbital elements. This caused the test orbits to miss the Earth as they moved away from the mean value. The density of meteor orbits was is taken as the number of observed orbits within a given D-criteria range.

```
Eta_Orbs.Pas      to  DD_Orbs.Dat
  to Eta_Vm.Pas   to  DD_Vm.Dat
  to Eta_Stab.Pas to  Graphics
```

```
program Eta_Orbs;                {Orbit cross sections, hit Earth}
    uses  OrbElems, OrbDefns, GenUtil, Crt, Dos;
{The program produces a set of orbit cross sections that can be used
 to investigate the local density enhancement around the eta Aquarid
 meteor stream.  The orbital elements are put out into the file
 DD_Orbs.Dat.  Two options are available for determining the
 heliocentric velocity at the point of detection.  Inclination,
 ecliptic longitude and speed.  Angles in  z x-y plane, x-z y plane
 and speed.}

program Eta_Vm;                  {Survey Orbital Densities}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the AllBox.eta or BigBox.eta ReducedData files.
 It produces a Right Ascension versus Declination diagram.  Those
 meteors which come within DD < 0.10 of the test orbit are marked on
 as larger spots.
 The idea of this mod (Eta_Vm) is to carry out cross sections in the
 orbital elements parameter phase space and check that the DCrit
 orbits provide a local maximum in the density of meteor orbits.  The
 test orbits used to provide the cross sections are found in
 DD_Orbs.Dat.  All of these test orbits intersect the Earth's orbit.
 The number of orbits within DD levels of 0.10, 0.07, 0.05, 0.03 are
 recorded in the file DD_tmp.dat.}

program Eta_Stab;                {Plot i, Vm, eclipt stability plots.}
    uses  OrbDists, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Stability profiles for the orbital elements given DCrit values.  The
 test orbits are from DD_Orbs.Dat and the DD data from DD_Vm.Dat.
 Programs used to obtain this data are Eta_Orbs.Pas, Eta_Vm.Pas and
```

```
 this program provides the graphical output.}

program Thes_DD;                {Orbital Densities varying elements}
    uses  OrbAstro, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the AllBox.eta ReducedData file and produces a
 Right Ascension versus Declination diagram.  Those meteors which come
 within DD < 0.10 of the new ADT eta Aquarid orbit are marked on as
 larger spots.  This mod (Thes_DD) searches across the orbital elements
 and check that the DCrit orbits provide a peak in the distribution at
 the mean orbit chosen to represent the eta Aquarid orbit.  Note that
 the orbits move away from an intersection with the Earth as a result.
 The output data of how many within DD < 0.10, 0.07, and 0.03 is
 written to DD_Stab.Dat.}

program Thes_Stb;               {Plot q e i w L stability plots.}
    uses  OrbDists, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Muck about file for thesis diagram.  Stability profiles for the
 orbital elements given DCrit values.  Here the density across the
 cross section is done by varying the orbital elements.  Note that
 this means the test orbits do not intersect the Earth's orbit across
 the cross section.  Plots the data from the file DD_Stab.dat}
```

A.12 Estimates of Measurement Uncertainties

This set of programs look at the effect of propagation errors that result from measurement uncertainties in the station data. The range of orbital elements, the implied D-Criteria associated with this and randomised distributions are all investigated.

```
program Err_Elem;               {Errors in orbital elements}
    uses   OrbElems, OrbReduc, OrbDefns, GenUtil, Crt;
{The timelags, elevation angle and time of observation can all be
 entered with complete freedom.  The program calls the GetOrbit routine
 in OrbElems and places the initial conditions and orbital elements
 into the file Reduc.Dat in the default directory.  This program is a
 modification of TOrbit.Pas to estimate propagation errors in the
 orbital elements.  These are evaluated in terms of the measurement
 uncertainties  in the station data.}

program Err_DD;                 {Compare element error to DD  value}
    uses  OrbAstro, OrbDefns, GenUtil, Crt, Dos;
{Uses the D-Criterion function DDcrit to estimate the DD value
 associated with the uncertainty in the orbital elements.}

program Err_Dist;               {Errors in element distributions}
    uses   OrbAstro, OrbDists, OrbElems, OrbReduc, OrbDefns,
           GenUtil, Graph, PostScript, Crt;
{The program generates a set of orbital distributions based on
 measurement uncertainties for a typical eta Aquarid meteor.  The
```

```
timelags, elevation angle and time of observation and azimuth angle
are all varied randomly within a normal distribution.  This
distribution reflects the measurement uncertainties of the station
data.  The program calls the GetAzimuthOrbit routine in OrbElems.
The counters are updated and the distributions displayed at the
end.  This program is a modification of TOrbit.Pas to estimate affect
of errors on the orbital element distributions.  The program generate
6000 randomised orbits based on a 1990 May 7, 0735:00 meteor
observation.  Plots of LongSun (implied by observation), 1/a, a, i,
w, q, V(helio) are all produced.}
```

A.13 Reduced Data Distributions

This suite of programs look at the reduced data stored on AMOR386. They do a cumulative count of the various reduced parameters stored in the ReducedData records. A number of selection criteria are programmed in order to look at the data in various ways. A menu is attached to choose which variable distribution is displayed. The programs do not plot one variable against another. I have found it necessary to use a more ad hoc one program per distribution to do that.

```
program Dist;                    {Reduced Data Distribution}
    uses  OrbDists, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the Orbit_<hr>. ReducedData files for one day.
 It then produces the required cumulative distribution diagrams. The
 interactive display section has an inbuilt help.  If double key
 strokes are required to select an option then they must be pressed
 within 2 second of each other to be useful.  This is a simple
 non-OrbView version of the distribution display programs.}

program DistDay;                 {Orbital Distributions for a Day}
    uses  OrbView, OrbAstro, OrbDists, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Reads in data form the Orbit_<hr>. ReducedData files.  It then
 produces the required cumulative distribution diagrams.  The
 interactive display section has an inbuilt help.  If double key
 strokes are required to select an option then they must be pressed
 within 2 second of each other to be useful.  The program selects the
 day to be viewed by menu.}

program DistLot;                 {Orbital Distributions, full data set.}
    uses  OrbAstro, OrbDists, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Reads in data form the Orbit_<hr>. ReducedData files.  It then
 produces the required cumulative distribution diagrams.  The
 interactive display section has an inbuilt help.  Provides a version
 of Dist.Pas which views all the data with full flexibility about
 choosing the various slices.}
```

```
program Scales;                    {Orbital Distribution Scales}
    uses  OrbDists, OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Provides a test bed in which to develop the graph layout for
 routines in the OrbDists unit.}

program DistPlay;                  {Reduced Data Distribution Diagrams}
    uses  OrbAstro, OrbDists, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Reads in data form the Orbit_<hr>. ReducedData files.  It then
 produces the required cumulative distribution diagrams.  The program
 has been modified from DistLot.Pas to provide a playground to try
 new comparison methods.}
```

A.14 Comparison Programs

```
program Geo_Diff;                  {Geometric vs Diffusion altitudes.}
    uses  OrbAstro, {OrbView,} OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Reads in data form the Orbit_**. ReducedData files and compares the
 geometric altitudes with those implied by the exponential decay
 curve.  A line is overlaid on the graph with geometric = decay
 altitudes.  A compiled version of the program with the menu selection
 enabled is held in C:\Useful.  Note that some days follow the
 'agreement' line well others do not.  Is this related to the phase
 calibration constant? }

program RA_Dec;                  {Plot R.A. Dec from Orbit_**. files}
    uses  OrbAstro, OrbView, OrbDefns, GenUtil,
          Graph, PostScript, Crt, Dos;
{Reads in data form the Orbit_**. ReducedData files and produces a
 graph of Right Ascension versus Declination.  An executable copy is
 held in C:\DATA\ Useful.}

program Thes_RAD;                  {Plot R.A. Dec from Orbit_**. files}
    uses  OrbDefns, GenUtil, Graph, PostScript, Crt, Dos;
{Reads in data from the Orbit_**. ReducedData files and produces a
 graph of Right Ascension versus Declination.  This version is a muck
 about file for a thesis diagram:
     shift RA range to 120 to 360 to 120 degrees.
     put on lines for the meridian at 0700hrs.}

program Zen_Az;                    {Plot Zenith angle vs Azimuth}
    uses {OrbView,} OrbDefns, GenUtil,Graph, PostScript, Crt;
{Reads in data from the Orbit_<hr> ReducedData files.  It then
 produces a graph of zenith angle versus Azimuth for the meteor
 observation.  This should give a plot of the effective meteor
 detection regions of the radar system.}
```

```
program Zen_Elev;                {Plot Zenith angle vs Elevation}
    uses {OrbView,} OrbDefns, GenUtil,Graph, PostScript, Crt;
{Reads in data from the Orbit_<hr> ReducedData files.  It then
 produces a graph of zenith angle versus Elevation angle for the
 meteor.  This will provide some information for correcting wind
 shears for non-vertical motion of the meteor.  Otherwise just
 for interest.}
```

A.15 General Programing

```
program TestPost;                {Test postscript output}
    uses  Graph, PostScript, Crt;
{Puts a set of symbols and lines on the screen so that they can be
 compared with the postscript output hard copy.  I used this while
 developing the PostScript ghost unit.}

program TestTrig;                {Test trig functions in OrbAstro}
    uses  OrbAstro, GenUtil, Crt;
```

Appendix B

Unit File Descriptions

Turbo Pascal allows programs to be broken up into several unit files called by an overall program. The unit files associated with the orbit radar are described in this appendix. Effectively they form a subroutine library. They also define a large number of variables and type declarations. The interface section of a unit includes declarations for all the constants, variables and routines available to a program that calls or uses that unit. The implementation section contains the actual code. I have included the interface section for all the AMOR units in this appendix. Actual source code is available on the VAX and PC for inspection. The code for routines directly involved in observing and reducing meteor orbits is contained in the source code appendix.

The observing code units are a second generation set started in October 1989 after some experience with running the orbit radar from the PC. They were copied from an original set Met* * *.Pas in a major reshuffle of the observing program. The only substantial change to this code since then is the establishment of a parent program to run the main observing program. The code units only slightly affected by this later change.

Contents

B.1 OrbDefns

```
unit OrbDefns;                    {Meteor Orbit Definitions}
interface
    uses Dos;
{Provides a definitions unit for the meteor orbit radar code.}

const FieldMax = 1800;
          {Number of data sweeps in AtoDMem while running the experiment under
           field conditions.  Outside the Turbo interactive environment.}
      TestMax = 400;
          {Data sweeps when running test routines inside Turbo environment.}
      MaxDetect = 48;
          {The number of positive detections that can be handled
           simultaneously.  A number this large allows for 6 simultaneous
           persistent meteors.  It allows DetectInfo to hold the subsequent
           redetections until the lifting routine has taken six observation
           records out of the AtoDMem.}
      BuffSize = 1280;
          {Number of bytes in an Observation record; ObsvBuff.}
      MaxStore = 240;
          {The number of meteor observations that can be stored in the
           extended memory storage space before stopping and writing to disk.}
      DMACard = 785;
          {Memory location of the I/O ports and DMA interface card, ($311) }
      Prf = 379;
          {Radar pulses per second.  Prf = 379 gives 6km range bins.}
      PersSweeps = Prf;
          {Number of sweeps per PersCount.  Equal to Prf makes the variable
           PersCount a timing counter in seconds.}

{Memory management and addressing for the AtoDMem space and MetStore.}
type  SweepBlock = Array[0..255] of byte;
      SweepPtr = ^SweepBlock;
      SweepPointers = Array[1..FieldMax] of SweepPtr;
      SweepAddr = record
                    Page : byte;          {Address for DMA page register}
                    MSB : byte;           {Start MSB of the DMA page}
                  end;
      SweepAddresses = Array[1..FieldMax] of SweepAddr;
      ExtAds = record
                 HiByte : byte;
                 LoWord : word;
               end;
      StoreAddresses = Array[1..MaxStore] of ExtAds;

{Meteor orbit observation record structures.}
type  DetectRecord = record
                       Number : byte;           {Save number of the echo}
                       Hour,Min,Sec : byte;     {Time (NZST), date added later}
                       Detected : word;         {Sweep number at detection}
                       SweepOffset : word;      {Index, hence range bin}
                       Threshold : byte;        {Detection threshold; Home}
                       PersTime : byte;         {Number of seconds active}
```

```
                    DMAHoldFlag : word;      {Sweep location of Hold Flag}
                  end;
    Vector = Array[1..250] of byte;
    SiteProfile = record
                    IndexOne : word;         {SweepNum of Profile[1] }
                    DetectSweep : word;      {SweepNum of detection}
                                             { = IndexOne if no detection}
                    RangeBin : byte;         {Range in 6km interval bins}
                    Threshold : byte;        {Detection threshold of site}
                    Profile : Vector;        {Meteor echo amplitude profile}
                  end;
    Observation = record
                    Year,Month,Day : byte;
                    Hour,Min,Sec : byte;     {Date and time in NZST}
                    Number : byte;           {Sequential number in Metstore}
                    PersTime : byte;         {Duration of persistence (sec)}
                    RangeOffset : byte;      {Home offset to peak (uS)}
                    N1,N2,N3 : byte;
                    Home,Nutt,Spit : SiteProfile;
                    Tin,Tos : Vector;
                  end;
    PersistentTimes = Array[1..MaxStore] of word;
    PhaseVector = Array[1..250] of 0..360;
    DateRecord = record
                   Year,Month,Day : byte;
                 end;
    HourValues = array [0..23] of integer;

{Data reduction record structures.}
type  SiteData = record          {total of 26 bytes in record.}
                 Noise : byte;          {Noise recorded by detection routine}
                 Max : byte;            {Maximum amplitude, smoothed }
                 Imax : byte;           {Index location of maximum }
                 NumHalfCycles:shortint;{Number of Fresnel half cycles}
                 Range : integer;       {Range to the meteor trail (km) }
                 Rise : byte;           {Echo profile rise time (pulses) }
                 Decay : byte;          {Exponential decay time (pulses) }
                 DiffHeight : real;     {Diffusion height  (km) }
                 Altitude : real;       {Geometric altitude (km) }
                 Velocity : real;       {Fresnel velocity (km.s-1) }
               end;
    ReducedData = record       {total of 298 bytes in record.}
                 Year,Month,Day : byte;
                 Hour,Min,Sec : byte;   {Date and time in NZST}
                 Number : byte;         {Sequential number in Metstore}
                 PersTime : byte;       {Duration of a persistent (sec)}

                 Home,Nutt,Spit : SiteData;
                 Lag12 : real;          {Lags between Home and Nutt sites }
                 Lag13 : real;          {  Home and Spit       (pulses)   }
                 Lag23 : real;          {  Nutt and Spit       (pulses)   }
                 AvTin,AvTos : real;    {Average of locked phase channels}
                 Phase : real;          {Relative phase angle, Home aerials}
                 Elev : real;           {Elevation angle of the meteor trail}
```

```
                AvFresVel : real;      {Average Fresnel velocity (km.s-1) }
                AtmosDecel : real;     {Fresnel deceleration (km.s-2)}

                Vx,Vy,Vz : real;       {Components of observed velocity}
                Azimuth : real;        {Azimuth angle of observed meteor }
                Zenith : real;         {Zenith angle of meteor trail }
                Speed : real;          {Calculated speed of the meteor }

                RightAsc : real;       {R.A. of the corrected radiant}
                Declin : real;         {Declination of corrected radiant}
                SpeedGeo : real;       {Corrected geocentric speed}

                DayNumber : integer;   {Day of the Year for observation}
                LongSun : real;        {Longitude of the sun at observation}
                Long1950 : real;       {Longitude at mean equinox of 1950.0}

                Rm : real;             {Heliocentric radius of velocity Vm }
                Vmx,Vmy,Vmz : real;    {Components of heliocentric velocity}
                SpeedHelio : real;     {Heliocentric speed of the meteoroid}
                LongAsc : real;        {Longitude of the ascending node}
                LongAsc1950 : real;    {  at mean equinox of 1950.0 }
                Inclin : real;         {Inclination of the orbit}
                SemiLatRect : real;    {Semi-latus rectum of meteor orbit}
                Perihel : real;        {Argument of perihelion (deg) }
                Ecc : real;            {Eccentricity of the orbit}
                SemiMajor : real;      {Semi-major axis (A.U.) }
                PeriDist : real;       {Perihelion distance (A.U.) }
                PeriLong : real;       {Ecliptic longitude and latitude }
                PeriLat : real;        {  of perihelion (radians) }
              end;


var   MaxSweep : word;
          {Sweeps of data in the AtoDMem space.  All the array stores are set
           up for the larger option of FieldMax.  Using TestMax will allow the
           test programs to be run inside the Turbo environment.}
      AtoDMem : SweepPointers;
          {Array of pointers for the handling program to access sweep data.}
      AtoDMemIRQ : SweepAddresses;
          {Addresses for the detection routine to reset the DMA registers.}
      SweepNum : word;          {Sweep index location within the AtoDMem.}
      DetectCount : Sweepblock;
          {The above threshold counter array to provide 'spike' suppression
           and positive identification of a meteor.}
      HomeThreshold : word;
          {Detection threshold for the Home site.  This is the only channel
           that the interrupt routines check for a meteor's presence.}
      NuttThreshold,SpitThreshold : word;
          {Detection thresholds for the other two channels.}
      DetectInfo : Array[1..MaxDetect] of DetectRecord;
          {Information recorded by the interrupt routine during the sweep the
           meteor was detected in.}
      DetectNum : byte;         {Array location of the last meteor detected.}
```

```
      NewNum : byte;          {Sequential number for new meteor detections.}
      ReleaseNum : byte;      {Last meteor released to the handling program.}
      LiftNum : byte;         {Meteor the handler is working on.}
          {ReleaseNum - LiftNum  gives the number of meteor echoes waiting to
           be 'lifted' out of the AtoDMem space.}
      ObsvBuff : Observation;
          {Data buffer for the temporary storage of meteor observations as
           they are lifted out of the AtoDMem space and prior to writing them
           up into extended memory.  27 May, 1989 : Removed the buffer
           ProcBuff.  I decided not to implement any more processing while the
           system is collecting and storing the observation records.  It takes
           too much extra space plus the complexity of having it running.}
      NeedToStore : boolean;
          {Flag that indicates to the detection interrupt routine when the
           observation buffer needs to be stored to extended memory.}
      ToExt : boolean;        {Direction of the extended memory move.}
      MetStore : StoreAddresses;
          {Precalculated addresses in extended memory for the temporary
           storage of meteor observation records.}
      PersCount : word;
          {A sequential count of seconds elapsed since the start of the run
           or since the last write to disk of the MetStore contents.}
      PersLow : word;         {Counts the sweeps in a PersCount interval.}
      PersInfo : PersistentTimes;
          {Records the length of time a persistent echo lasts for.  The
           number gives the time interval in PersCount units (seconds).}
      SaveNum : byte;         {Meteor being written to extended memory.}
      ProcNum : byte;         {Observation to be read from the store.}
      DateStart,DateEnd : DateRecord;
          {The start and end dates of the time that the detection routine is
           active.  That is the date at SaveNum = 0 and after SaveNum =
           MaxStore.  This allows for the midnight rollover of the data.}

      AlarmCount : word;      {Counts seconds since last meteor echo.}
      Comment : string;       {Reason for stopping the observing run.}
      IntSave0B : pointer;    {Default pointer for IRQ#3 servicing  (IRET).}
      IntrShutDown : pointer; {Pointer to the code to shut down a run.}
      IntrDetector : pointer; {Pointer to the detection isr.}
      IRQCounter : longint;   {Number of interrupts processed.}

      StopRun : boolean;
      Answer : char;

      NoiseStr : string;        {for graphics display}
      Regs : registers;
      i,j : integer;
      Hour,Min,Sec : byte;
      Finished : boolean;       {debugging control variable.}
      GotOne : boolean;         {another !!}
implementation
begin  end.
```

B.2 OrbRadar

```
unit OrbRadar;                    {Meteor Orbit Radar.}
interface
    uses  OrbDefns, GenUtil, Dos, Crt;

{A code unit for procedures used while running the Meteor Radar with the
 'luggable' AT.  SetAtoDMem needs to be called to set up the heap to receive
 the raw amplitude data from the receiver rack.  MaxSweep must be initialised
 before this is done.  The interrupt detection routine is included in this
 unit.  Initialisation code installs OrbitExit as an exit procedure.  It also
 sets variables to default values applicable to the meteor orbit observing
 program.}

{$F+}    {Force the far call mode to subroutines so Detector
          can easily be used as interrupt service routine.}

{Andrew Taylor,   20 October  1989.}

var   OrbitExitSave : pointer;     {Save pointer to unit exit code.}


procedure SetAtoDMemory
    (var AtoDMem:SweepPointers; var AtoDMemIRQ:SweepAddresses);
    {Allocates enough memory on the stack for MaxSweep sweeps.  The routine
     first shifts the base of the heap so that blocks of data from one sweep
     never cross a page boundary.  The DMA controller has a 4-bit page address
     which must be reset to cross the boundary.  This is done in the IRQ
     routine for each new sweep of data.  MaxSweep must be initialised before
     the routine is called.  If the AtoDMemory is being set up to run under
     field conditions the routine checks that there is at least 4 MBytes of
     disk space free to store the observations.}

function CheckMemSizes : boolean;
    {The routine reads off the memory sizes of the various code, data, stack
     and heap segments.  The program can then be terminated by pressing the
     space bar without having to actually run the radar.  The call to this
     routine is left as a comment in the Observe.Pas program.  The memory
     addresses need to be evaluated by a procedure rather than with the
     debugger since the full AtoDMem and the turbo interactive environment
     cannot both be resident at the same time.}

procedure LookAtSweepBlocks;
    {An environment to display the 256 bytes associated with any sweep number.
     Commands are:  quit (q), display sweep (number),
                    next sweep (any thing else and a return). }

procedure LookAtDetectCount
    (SweepNum:word);
    {Dumps the current contents of the array DetectCount onto the screen.
     If the variable SweepNum is not passed as a parameter it retains the last
     value used.  The DetectCount array will be in the state appropriate for
     having just viewed sweep number SweepNum - 1.}
```

```
procedure LookAtDetectInfo;
    {Writes out the current contents of the detection data array DetectInfo.
     Press return to see the next page and again when finished.}

procedure ResetVars
    (AtoDMem:SweepPointers; var DetectCount:SweepBlock;
     var DetectNum,NewNum,ReleaseNum,LiftNum,SaveNum:byte;
     var PersInfo:PersistentTimes; var PersCount,PersLow:word;
     var SweepNum,AlarmCount:word; var StopRun,RunStopped:boolean);
    {Resets the DMAHoldFlag locations, control variables and index numbers
     prior to restarting the transmitter pulses.}

{Interrupt service routines included in the unit.

 ShutDown
     Stops servicing the DMA controller so that no more data is read in.  The
     routine continues to be called until all the meteor detections that are
     waiting to be processed have been lifted out of the AtoDMem data.  Once
     that is done and they have been shifted to the MetStore the IRQ#3's can
     be masked out by the main control program.  This routine does not mask
     IRQ#3, re-enable the timer (IRQ#0) or switch the TX trigger pulses off.
     It allows the main control program to decide on this.  If the TX stops
     pulsing the HT voltage seems to creep up.  If the transmitter is feeling
     fickle it will stop!!  Switching the TX off makes diagnosing any problem
     more difficult.

 Detector
     Provides the meteor detection routine for the orbit radar.  Services the
     DMA data acquisition.  Looks at the data collected in the last sweep to
     check whether a meteor is present.  Detection requires 6 successive
     sweeps above the threshold level.  If the number of times the signal must
     be above threshold for a positive detection is changed within the case
     statement then DetectCount[i] should be reinitialised as one more.  A new
     detection will be assumed to be from a persistent trail if it occurs
     within 2 sec (0-2 of PersCount) of a previous detection at a similar
     range bin.  The routine also synchronises the move to extended memory of
     a completed observation record.  If any changes are made the range
     checking should be switched on until the code is thoroughly debugged.
     Alarm conditions which shut down the run are :

     DMA data overrun : When the lifting routine has not finished lifting an
             observation out of the AtoDMem before the routine wants to reset
             the DMA controller to overwrite the sweep data.

     DetectInfo overrun : The main control program has not had time to process
             the data currently stored in the DetectInfo[DetectNum] record
             before it is over written by the new detection.

     Lack of detections : When there has been no meteor echoes detected for
             the previous fifteen minutes.  The transmitter has probably
             stopped for some reason.  A fifteen minute gap between detectable
             meteors is most unlikely.  During the early evening there are
             often gaps of many minutes.  In the early morning these gaps are
             seldom more than a minute long.
```

```
     DetectCount has an illegal value : One of the DetectCount array values
            has gone out of range.  The program crashes out.  This provides
            some very weak range checking on the detection routine.
}

procedure PauseForNoise
    (var ObsvBuff:Observation; var SaveNum:byte; var NoiseTooHigh:boolean);
    {If more than 60% of the amplitudes are above threshold when a noise
     sample is taken then the noise level is considered too high.  It could
     be caused by a burst of interference in which case waiting for a short
     period of time then restarting should be sufficient.  Occasionally the
     base level on the FM links drifts too high.  The observer needs to check
     and adjust them in this case.  Sporadic-E echoes also triggers this
     condition.  The only realistic solution is to stop and wait for it to go
     away.  The routine triggers an alarm and waits.  If the observer is
     absent it will return and attempt a restart after 2 minutes.  Since the
     last observation is the one affected by the high noise level it is not
     worth saving.  SaveNum is modified so that WriteOutStore does not store
     the noisy observation to disk.  The ObsvBuff.N# values are set to zero
     to ensure the program does not retrigger off the same samples.}

procedure CheckDiskFree;
    {Checks that there is enough disk space free on drive C: to store a full
     MetStore of 240 observations.  If there is less than 300 KBytes this
     will generate a Halt(4) and terminate the run.}

procedure OrbitExit;
    {Exit procedure for error conditions in the OrbRadar.Pas unit.  The error
     conditions noted above are all covered.  The errors are handled under a
     Halt(1) function (Exit = 1).  Previously if the noise level on any of
     the RX channels gets too high the program would terminate with Halt(2).
     Generally the solution is to wait a while and start again.  If the
     observer is absent then it is better to get the program to do it itself.
     This is now handled by the PauseForNoise procedure,  3 April 1990.}
```

B.3 OrbLift

```
unit OrbLift;                    {Lift Meteor Orbit data from AtoDMem.}
interface
    uses OrbRadar, OrbDefns, GenUtil, Dos, Crt;

{Using data obtained by the detection routine this unit provides code to:
 Lift the meteor echo profiles out of the AtoDMem,  move observation buffers
 to and from the extended memory store,  add the date, register the duration
 of persistent echoes and then write the meteor observations to disk.  The
 procedure SetMetStore needs to be run to set the address locations before
 any use is made of the extended memory.  The initialisation code installs
 LiftExit as an exit procedure.  The global descriptor table outlines are
 initialised with SetGDTbones.}

{$F+}    {Force the far call mode to subroutines so ExtMemMov
          can be used easily as interrupt service routine.}
```

```
{Andrew Taylor,   20 October  1989.}

type  LocalDescTable =  record
                          SegLimit : word;     {Segment limit, 1 - 65536 bytes}
                          LoWord : word;       {24-bit physcial address,     }
                          HiByte : byte;       {     0 to 16M - 1.           }
                          DataAccess : byte;   {Access rights  = $93, CPLO-R/W}
                          Reserved : word;     {Reserved word, must be zero }
                        end;
      GlobalDescTable = record
                          Dummy   : LocalDescTable;      {preset to zero }
                          Local   : LocalDescTable;      {preset to zero }
                          Source  : LocalDescTable;      {Source to be moved }
                          Target  : LocalDescTable;      {Destination of data }
                          BiosCS  : LocalDescTable;      {preset to zero }
                          Stack   : LocalDescTable;      {preset to zero }
                        end;

var  IntSave60 : pointer;
     IntrExtMemMove : pointer;
     GDTread,GDTwrite : GlobalDescTable;
     LiftExitSave : pointer;      {Save pointer for exit code.}


procedure LiftMeteorObsv
    (LiftNum:byte; var ObsvBuff:Observation);
    {Lifts the data for one meteor observation out of the AtoDMem space and
     stores it as an Observation record in the buffer ObsvBuff.  LiftNum
     refers to the array of detection information (DetectInfo) noted by the
     interrupt detection routine.  If the routine does not find a meteor
     profile on the Nutt or Spit sites then an amplitude profile is taken from
     the most probable location of an echo.   I was not intending to save an
     observation if the lifting routine could not make a positive detection on
     the other two sites.  Saving everything does not tie up much more space.
     It also helps decide how often the different sites do not get common
     echoes.  Collecting these sorts of statistics is easier as a separate
     exercise.  The routine assumes detection on the Home site channel.}

function FindProfile
    (StartSweep:word; OffsBase:word; var Offs:word; var Site:SiteProfile)
       : boolean;
    {Starting at StartSweep this routine runs a detection routine on one of
     the two sites not already processed by the interrupt detection routine.
     It searches five range bins above the range bin the Home site echo was
     detected in.  If the remote sites are shifted this search pattern may
     need to be changed.  (The Spit receiver aerial was moved April 1989
     necessitating the change to 5 range bins).  OffsBase should be the start
     offset for the search in that site profile.  Offs is the location of a
     positive detection.  Other detection information is in the Site record.}

procedure LiftProfile
    (var Site:SiteProfile; Offs:byte);
    {Looks into the AtoDMem space and picks out the echo amplitude profile
```

```
     with a quarter, half, quarter average.  This average is centred on the
     offset, Offs in the sweep block.}

procedure RangeScanOffset
    (var Site:SiteProfile; Offs:byte; var ROffs:byte);
    {Does a three point fit on the adjacent rangebin amplitudes to estimate
     the offset of the peak from the location of the rangebin with the
     maximum amplitude.  The answer ROffs, in microseconds, is positive
     toward greater ranges.  It is necessary to have the quarter, half,
     quarter profile available to make this calculation.}

procedure LiftPhase
    (var Site:SiteProfile; var Phase:Vector; Offs:byte);
    {Picks out the phase channel vector from the same range bin and starting
     sweep as the Home site.  Currently it is a direct lift of the data in
     that range bin.  A quarter, half, quarter average as with the amplitude
     profiles may be more appropriate (28 Feb, 1989). Offs locates the phase
     data associated with the required range bin.}

procedure LiftNoise
    (var Site:SiteProfile; Offs:byte);
    {Takes an amplitude sample from the channel when no echo was detected.
     I cannot see why I have used a separate routine to the LiftProfile one
     above for the case where there FindProfile cannot find an echo profile.}

procedure NoiseSample
    (Threshold,initOffs:byte; StartSweep:word; var Noise:byte);
    {Takes a noise sample from data recorded when the meteor was detected at
     the Home site.  initOffs gives the offset required to identify which site
     the sample is being taken from.  The routine sums the amplitudes below
     the upper <SITE>Threshold for 8 sweeps giving a noise average.  There
     are 8*48 possible sample locations.  If less than 40 percent of the
     values sampled are below the threshold of 30 then a noise value of 255 is
     returned.  This should trigger an alarm condition to the observer.}


procedure SetMetStore
    (var MetStore:StoreAddresses);
    {Calculates the addresses in extended memory at which meteor observations
     can be stored.  A maximum of MaxStore observations can be stored there.
     Currently MaxStore = 240 requiring 300K bytes of extended memory.  The
     extra space (86K bytes) can be used to store more fully reduced data.}


procedure InitGDT
    (var GDT:GlobalDescTable);
    {Sets up the features of the Global Descriptor Table common to both read
     and write operations with extended memory.}

procedure SetGDTbones
    (var GDTread,GDTwrite:GlobalDescTable);
    {Provides skeletons for the Global Descriptor Tables used by the BIOS
     interrupt $15 function $87.  It is established prior to use to speed up
     the read from and write to extended memory routines.  The transfer
```

```
     routine need only put the extended memory addresses in.  The transfer is
     always to or from the record ObsvBuff.}

{ExtMemMov : interrupt service routine
     Moves observation data to and from the storage space in the extended
     memory.  The routine makes use of the BIOS interrupt service $15
     subfunction $87.  The extended memory locations are all precalculated
     and reside in the array MetStore.  The boolean ToExt determines the
     direction of the move and should be set before this routine is called.
     The routine is usually installed as interrupt service $60.
     For ToExt
       true  : Takes the data from the ObsvBuff and places it in the
               storage space.  It uses GDTwrite.
       false : Moves the data from the observation store into ObsvBuff
               using the precalculated GDTread array.}


procedure WriteOutStore
    (SaveNum:byte; DateStart,DateEnd:DateRecord;
     var PersInfo:PersistentTimes);
    {Reads the meteor observation records in the extended memory store.  The
     date of observation and the duration of a persistent echo (PersTime) are
     added to the record.  The routine finally writes the observation out to
     the appropriate disk file.  All data is placed in a subdirectory of the
     \DATA root directory.  The relevant subdirectory is located by day and
     the observation record goes in the pertinent NZST hour file. The
     format is        C:\DATA\ <yymmdd> \ NZST_<hh>
     Interrupt channel IRQ#0 needs to be running to service the disk
     operations.  Interrupt $60 must be set to point at ExtMemMove.  }


procedure LiftExit;
    {Exit procedure to ensure that most of the data in the MetStore is saved
     should a runtime error terminate the program.  Resets the interrupt
     vectors and the interrupt controller then calls WriteOutstore.  If the
     observing routine terminates with a runtime error then an alarm is set
     off.  An error with the BIOS Block Move is handled under Halt(3).  An
     attempt is made in this case to read down the contents of the MetStore
     and write them out to disk.}
```

B.4 OrbReduc

```
unit OrbReduc;                    {Reduction of meteor orbit observations}
interface
uses   OrbGraph, OrbDefns, GenUtil, Graph, PostScript, Crt;

{This unit contains the procedures to screen out incomplete observations,
 provide the necessary data for orbit calculations and provide profile
 information for meteor aeronomy.

 The constant ReducSweep needs to be set to the value of MaxSweep that was
 being used by the observing program when the data was collected.  The
 reduction program will need to be recompiled to deal with a new value.}
```

```
const  SatHome = 200;                    {Receiver amplitudes above these }
       SatNutt = 240;                    {  values are within about 90%   }
       SatSpit = 230;                    {  of RX saturation.             }
       EarthsRadius = 6468;              {Earth's radius at 44 South (km) }
       TopTin = 184;                     {Top and bottom cutoffs to use Tos }
       BotTin = 68;                      {  to calculate angle (AtoD amp)   }
       HomeDelay = 25;                   {Time delay from TX trigger til }
       NuttDelay = 40;                   {  arrival of ground pulse (km) }
       SpitDelay = 37;
       TooHigh = 120;                    {Altitude ceiling and floor to  }
       TooLow = 70;                      {  resolve phase ambiguities (km) }
       AerialSpacing = 4.96;             {Home site spacing (wavelengths) }
       WaveLength = 11.45;               {Radar wave length (m) }
       PointsInFit = 16;                 {Pulses of Fresnel sub domain }
       ReducSweep = FieldMax;            {Sweeps in AtoDMem at Observation}

type   IntVector = Array[1..250] of integer;
       RealVector = array[1..250] of real;
       HalfCycleSizes = array[0..60] of byte;

var    SmoothBuff : Observation;
       Error : word;                     {Error indication variable}
       Comment : string;                 {Description of error type}
       TestGraphicsOn : boolean;         {Switch for graphical test display}
       PhaseCalib : HourValues;          {Calibration constant array}
       SdevTin,SdevTos : real;
       PhasDiff,ElevDiff : real;



procedure WriteReducedData
    (var Buffer:ReducedData; FileName:string);
    {Writes the data out to the text file called FileName.  To write to the
     screen just use the FileName:= ''.  Puts out the information contained
     in the ReducedData records.  This includes the information calculated
     within the OrbReduc, OrbLags, OrbFres and OrbElems units.  It largely
     supercedes the WriteOrbitBuffer procedure in OrbElems.}

procedure WriteAeronomyData
    (var Buffer:ReducedData; FileName:string);
    {Writes the data out to the text file called FileName.  To write to the
     screen just use the FileName:= ''.  Just writes out the aeronomy data
     from the ReducedData record.  Writes out the Home site data and the
     phase and elevation information.}

procedure TransferObsvData
    (var ObsvBuff:Observation; var Buffer:ReducedData);
    {Puts the data from the observation record into a ReducedData record.}

procedure TriSmooth
    (var Site,SmoothSite:SiteProfile; R:byte);
    {This procedure carries out a triangular smoothing to a range of R
     elements on either side of the principal value.  The smoothing leaves
```

```
     the same total amplitude sum in SmoothSite.Profile as in Site.Profile.
     The SmoothBuff profiles in this reduction package mostly use R = 3.}

procedure CompareProfiles
    (var MeteorObsv:Observation; var ProTop,ProMid:SiteProfile; Noise:byte);
    {Displays two echo amplitude profiles.  The noise is that of the 'raw'
     profile usually in ProTop.}

procedure FindMax
    (var Site:SiteProfile; var Imax,Max:byte; LimitIndex:byte);
    {Finds magnitude and location of the maximum amplitude in the profile.
     The parameter LimitIndex restricts the range in which the maximum is
     searched for.  Set it to 250 if the whole profile needs to be searched.
     The routine starts searching for the maximum at index 10.  This ensures
     that it does not run into problems with the despiking routine not
     coping well with the initial few points.}

procedure DifferentiateProfile
    (var Amp:SiteProfile; Start,Finish:integer; var AmpDash:IntVector);
    {Takes the amplitude profile in Amp.Profile and differentiates it
     between the two bounds Start and Finish.  For the MaxRiseLags method
     these points are from index 16 through to 5 pulses after the maximum
     amplitude.  If things are reasonably stable the result should remain
     positive.  To reduce processing AmpDash[i] values are a factor 60*w,
     (w = 5) larger than the actual value of the differential.  Start
     should be >= 16 and Finish <= 235.}

procedure DrawDiffProfile
    (var Amp:SiteProfile; Start,Finish:integer; var AmpDash:IntVector);
    {Plots the differential vector AmpDash above the amplitude profile
     on an expanded time base.  This provides a diagnostic check for the
     differentiating routine.}

procedure GetDecay
    (var Site,Smoothed:SiteProfile; Imax,Noise:integer;
     var Decay:byte; var Error:word);
    {The procedure uses the newly developed differentiating routine to
     better identify the limits to use in making the decay fit.  It starts
     5 sweeps before the maximum decay slope and goes to the first
     (noise + 5) amplitude after the zero slope following the maximum decay.
     Where a significant rising edge is detected later in the profile the
     decay estimate is abandoned.  Either a positive 0.25 times the decay
     slope is found or a rising edge with the equivalent of 8 consecutive
     sweeps with a positive slope of 1.  11 Oct 1990.  The routine replaces
     the Get_Shape routine below.  The rise time is estimated in the
     MaxRiseSlopeLags routine.  Rise is the number of sweeps needed to rise
     from Noise to Max amplitude given the maximum rising slope.}


procedure MeanTinTos
    (Max:byte; var SmoothHome:SiteProfile; var VT:Vector;
     var Average,StdDev:real);
    {To find the mean and std. dev. of Tin or Tos.  We want at least 7
     values for a good average.  The best results seem to be obtained by
```

```
     starting this average at index 45 (about where the detection routine
     implicitly started to detect the meteor). This seems to correspond
     to the best lock on the Tin and Tos channels from which to calculate
     the elevation. If you go past about i = 80 - 120 in the profile,
     wind effects sometimes cause a shift in the phase.
     18 July, 1990 : With big meteor echoes the phase angle moves as the
     meteor moves toward the specular reflection point. In this case do not
     start the average until 70% of the maximum is reached.
     StdDev is the standard error of the mean.}

procedure ReadPhaseCalib
    (Year,Month,Day:byte; var PhaseCalib:HourValues);
    {The routine reads in the phase calibration data recorded in the data file
     PhaseCal.Dat. This provides an initialisation routine for GetPhase.}

procedure FitPhase_1983
    (AvTin,AvTos:real; var Phas:real);
    {Finds the phase difference between the two home receivers from the TIN
     and TOS averages. The code is based on the 9 November 1983 calibration
     of the Phase Channels. }

procedure OldFitPhase_1983
    (AvTin,AvTos:real; var Phas:real);
    {Finds the phase difference between the two home receivers from the TIN
     and TOS averages. The code is based on the 9 November 1983 calibration
     of the Phase Channels. This is the old routine that has holes in the
     phase angle distributions. }

procedure Get_Phase
    (Year,Month,Day,Hour:byte; var PhaseCalib:HourValues;
     AvTin,SdevTin,AvTos,SdevTos:real; var Phas,PhasDiff:real);
    {Finds the phase difference between the two home receivers from the TIN
     and TOS averages using FitPhase_1983. It then adds on the corrections
     for the phase calibration angle and the inverted data after June 1989.
     For the uncertainty in the phase I have just calculated the difference
     if the average Tin or Tos values were 2 standard deviations larger or
     smaller. This does not necessarily reflect the true experimental
     uncertainty especially around the phase values where I switch the
     empirical fit from one average value to another. To make the reduction
     package more robust the routine ReadPhaseCalib sets up the phase constant
     information from a data file (PhaseCal.Dat). Major problems like the
     1990 May 3,4 phase inversion need to be programed in here still. This
     will ensure that any reprocessed data is dealt with sensibly. Any new
     transmission line lengths or accidental swapping of the cables will mean
     another few lines need to be added to PhaseCal.Dat.}

procedure Get_Range
    (RangeBin:byte; var Range:real);
    {Finds the range to the Home site reflection point (on the meteor trail).
     This procedure is useful in establishing the range for comparisons of
     raw data on the Phase Range diagram.}

procedure Get_Ranges
    (var ObsvBuff:Observation; var Buffer:ReducedData);
```

```
    {Finds the range to the reflection points on the meteor trail).}

procedure Get_Elevation
    (Phas:real; var Elev:real);
    {Takes the calculated phase angle and the first estimate of the range and
     calculates the elevation and the altitude of the Home site reflection
     point.  Depending on how I resolve the range ambiguities will determine
     whether the range needs to be a variable in this routine.
     23 June  1989 : The spacing between the aerials is not 4.77 wavelengths
     but measures 56.8 metres.  It should be (56.8*26.2e6/3e8) or
     4.96 wavelengths.  This will have been affecting the earlier elevations
     and geometric altitudes.  I declared the spacing as a constant for
     the unit. }

procedure Get_Altitude
    (Elev,Range:real; var Altitude:real);
    {Establishes the altitude given a curved Earth geometry.}

procedure PlotASymbol(Symbol:SymbolType; Range,Phase:real; Size:integer);
    {Adjusts the scale to plot on a data point.  Initially range bin / phase
     and puts a cross on.}

procedure ResolveAmbiguities
    (Phas:real; var Elev,Range,Altitude:real; var Ambig:boolean);
    {Resolves most phase ambiguities on the basis of the range phase
     distribution.  It is difficult to tell if echoes at high elevations
     come from the north or the south.  Since these echoes are at closer
     ranges they could easily come from behind (north).}

procedure ResolvePhase
    (Phas,Lag23:real; var Elev:real; var Range:integer; var Altitude:real;
     var Ambig:boolean);
    {Resolves phase ambiguities by considering the time lag information and
     the range phase distribution.  The time lag, Lag23 identifies whether
     the meteor is travelling from the south or north.  Lag23 is negative
     where a meteor comes from the north (hence an echo returns from the
     south).  That is the meteor is detected at the Nutt site 2 before the
     Spit site receiver 3.}

procedure DrawDecay(Istart,Iend:integer; Amp,Tau:real; Noise:integer);
    {Draws on an exponential decay curve with a decay constant tau.  This
     routine has been modified (11 Oct 1990) to place the curve on the
     graph associated with the differentiation routine and GetDecay.}

function CheckOut
    (var ObsvBuff:Observation) : boolean;
    {Checks that the observing code detected a meteor on all three
     channels.  Returns false if either of the remote sites had a 'nil'
     detection.  Since the primary detection is done at the Home site
     this will always report a positive detection.  Other simple rejection
     criteria might be introduced into this routine.  This check function
     is run under ScrnObsv on the VAX, 1990 Oct 26.
     1991 Jan 10 : Added in a check that the RangeBin values for the Home,
     Nutt and Spit sites are non-zero.  901223 2047:11 #17 had Spit RB = 0}
```

```
procedure DeSpikeProfile
    (var V:Vector; Noise:integer);
    {De-spikes the amplitude profile.  This routine will remove noise spikes
     that last for only one pulse.  It does not provide as good a check on the
     last two amplitudes in the profile.  Any amplitude sample should be
     within a reasonable range of the average of the two adjacent samples.
     The routine uses 30 or 3*Noise, whichever is greater, as its bound.}

procedure WriteBuffer_Txt
    (var OutText:text; var Buffer:ReducedData);
    {Writes out the contents of the ReducedData buffer out to a text file.
     The routine adds a few '*'s between each reduced data record.  When the
     data text file is read this will allow a check to ensure that none of
     the numerical values have not overrun their fields.}

procedure ReadBuffer_Txt
    (var InText:text; var Buffer:ReducedData);
    {Reads in the contents of a reduced data record from a text file written
     with the WriteBuffer_Txt procedure.}
```

B.5 OrbLags

```
unit OrbLags;                    {Time Lags calculations}
interface
uses   OrbReduc, OrbGraph, OrbDefns, GenUtil, Graph, PostScript, Crt;

{This unit contains the procedures necessary to calculate timelags between
 the three receiver stations.  It is one of the sub units broken off from
 OrbReduc.Pas when that unit got too large, 14 April 1990.  I am adding a
 new set of timelags determining methods to this unit.  These might give a
 better answer than the XCorr approach.  They will provide a basis of
 deciding whether the profiles are any good.

 13 October 1990 : After a very busy week of programing (Rise Points, Max
 Slope and differentiation decay constants) I am quite sold on the Max Rise
 Slope time lag determinations.  I have been running comparisons between
 all four methods (including Rising edge and full profile cross correlation)
 and overall prefer the max slope method.  Essentially all of the observations
 where Rise points disagrees I put to errors in that routine.  The data
 collected after the new detection scheme was introduced has greatly improved
 the success rate of the max slope method.

 Max slopes is less CPU intensive and rejects profiles where wind seems
 to be affecting the rising edge very well.  It may also reject overdense
 trails with a slow rise to maximum.  There will be few of these at our
 sort of magnitude limit.  I hope.}


type   RiseSpotsData = record
                         Lag : real;
                         Diff : real;
```

```
                    end;
      RiseLagsArray = array[5..8] of RiseSpotsData;

procedure BaseShift
    (Noise:byte; var Site:SiteProfile; var Echo:IntVector);
    {Shifts the baseline up to the quiet level (noise) for XCorr calculation.
     It then calculates the average amplitude and moves the baseline up to
     there.  This does a good job of sharpening up the CCF.}

procedure GetRiseSection
    (var Site,Smooth,RiseOnly:SiteProfile);
    {This routine selects the rising edge of the profile and places just
     that piece into the RiseOnly profile.  The other amplitudes are filled
     with zero.  This will provide a profile that is less affected by the
     atmospheric effects of a long duration ionisation trail.  The smoothed
     profile is used to find the point where the rising edge reaches 80% of
     the maximum.  It considers the 40 amplitudes up to this point as the
     rising edge.}

procedure Shift
    (var V,VS:IntVector; s:integer);
    {VS is the vector V shifted right by s positions.  The routine does not
     wrap the profiles around.  It fills with zero.  This has essentially
     no effect on short duration echoes but makes the CCF sharper and more
     symmetric for longer echoes without the full decay curve.  The time lags
     seem to be correspondingly improved in these cases.}

procedure DrawCCFScales;
    {Draws up a set of scales for the CCF function.}

procedure PlotPointCCF(Pulse:integer; CCFAmp:real);
    {Plots a point on the DrawCCFScales graph.}

function LappLag
    (var Site1,Site2:SiteProfile) : integer;
    {Finds the time difference in pulses between the detection points of the
     recorded echo profiles.  If data is collected with a different value of
     MaxSweep then the constant ReducSweep should be changed for processing.
     This is found in OrbReduc.Pas.}

procedure XCorr
    (var Site1,Site2:SiteProfile; var AA,BB:IntVector;
     var PFit:real; var Error:word);
    {Takes the two meteor amplitude echoes and finds the time shift between
     them using a cross correlation procedure.  Before I adjusted the
     detection level on the remote sites the observing program would often
     not detect a meteor profile but would accidently pick one up.  Using
     manual sorting these were included in the *.Orb files.  Where the
     observing program has detected a meteor profile on that channel,
     the routine uses a narrow range for the cross correlation.  Where it did
     not detect a profile but a small amplitude echo has been included in the
     default profile the range of the cross correlation is increased.  To give
     a good view of the CCF the range is set at -50 to 50 for graphical
     testing.  Now with the new detection levels there is never any need for
```

```
     the wider CCF.  It is left in to allow reprocessing of the old
     observations.  In fact throwing away cross correlations that get too
     different from the detection time lags is a good idea.}

procedure DisplayProfiles
    (var ObsvBuff:Observation);
    {Puts out an amplitude display for the three echo profiles.}

procedure TimeLags
    (RisingEdge:boolean; var ObsvBuff,SmoothBuff:Observation;
     var Lag12,Lag13,Lag23:real; var Error:word);
    {Makes a copy of ObsvBuff as LagBuff.  Where the amplitudes of the echo
     profile are small, the cross correlations are done on the full profile.
     Where they have larger maximums, just the rising edges are used to try
     and eliminate some of the atmospheric effects on the ionisation train.
     The rising edge is more closely related to the motion of the meteor
     through the specular reflection point so using this section is
     preferable.  In either case the full profile or the rising edge only
     profile is transferred to the observation record LagBuff before the rest
     of the timelag processing is done.  The routine establishes the quiet
     level as the absolute zero.  The mean level of each profile is
     subtracted.  There is no need to run a window procedure since the new
     system collects a full 250 point profile from the relevant range no
     matter what.  The cross correlation routine is then run on the three
     profiles.  With TestGraphicsOn the routine will produce a series of
     profiles actually used for the cross correlation and display the CCF.}

procedure RisePoints
    (var Smoothed:SiteProfile; Imax,Max:integer; var Error:word;
     var Max50,Max60,Max70,Max80:real);
    {Using the smoothed echo profile the routine finds the sweeps on the
     rising edge where the amplitude rises to 50%, 60%, 70% and 80% of the
     maximum value.  Where any of these points are found at index less than
     30 an error condition is triggered.}

function CheckRiseLagsAgree
    (var RiseLags:RiseLagsArray; LappLag:integer; var RiseLag:real;
     var NumAgree:integer) : boolean;
    {The function takes the individual lags calculated from the four points
     and checks to see that they agree.  Where at least three agree to within
     +- 2 sweeps the time lag is accepted.  The mean is calculated
     accordingly.  NumAgree will have values of 4,3 when CheckRiseLagsAgree
     is true and zero when the result is false.}

procedure RisePointLags
    (var SmoothBuff:Observation; var Buffer:ReducedData;
     var Rise12,Rise13,Rise23:real; var Error:word);
    {Calulates the timelags based on points identified on the rising edge of
     the profile.  The points are found using RisePoints.  The timelags for
     each of the spots are compared.  Where three agree reasonably well the
     timelags Rise** are accepted.}

procedure FindMaxRise
    (Start,Finish:integer; var AmpDash:IntVector; var MaxDash:integer;
```

```
     var MaxRise:real; Noise,Max:byte; var RiseTime:byte; var Error:word);
    {Uses a parabolic fit to find the sweep index with the maximum rising
     slope, MaxRise.  MaxDash gives the maximum rising slope in the range
     Start to Finish.  If the maximum is too close to the boundaries then
     the routine cannot do a parabolic fit (MaxRise = 999.9).  The width
     of the maximum peak in AmpDash is the width at 70% MaxDash.  The
     average of these extreme 70% points must be within 6 sweeps of the
     parabolic fit MaxRise.  Otherwise the differential is said to be
     double peaked (MaxRise = 888.8).  Errors in this routine return
     Error = 17.  The rise time gives the number of pulses it takes to go
     from Noise to Max amplitude given the maximum slope (MaxDash/300).}

procedure MaxRiseSlopeLags
    (var SmoothBuff:Observation; var Buffer:ReducedData;
     var Max12,Max13,Max23:real; var Error:word);
    {Finds the point with the greatest slope on the rising edge for each
     profile.  These are compared to give the time lags between the three
     receiver stations.  The rise times of the three profiles are calculated
     and stored in Buffer.<Site>.Rise .}
```

B.6 OrbFres

```
unit OrbFres;                    {Velocity from Fresnel Oscillations}
interface
uses   OrbLags, OrbReduc, OrbGraph, OrbDefns,
       GenUtil, Graph, PostScript, Crt;

{This unit contains the procedures necessary to calculate the velocity based
 on the fresnel oscillations of the echo profile.  It is one of the sub units
 broken off from OrbReduc.Pas when that unit got too large, 14 April 1990.}

procedure DrawShapeScales;
    {Draws up a set of scales to display the amplitude profile of the Fresnel
     Oscillations.  It is used by the family of test programs Fres*.Pas.}

function PlotShapeY
    (Amp:integer) : integer;
    {Returns the y pixel of a given amplitude to be plotted on the
     DrawShapeScales graph.}

procedure SketchMeanAmp
    (Imax,Max:byte; var SmoothRaw,MeanAmp:SiteProfile);
    {On the first pass the routine searches through and finds the crossing
     points of the fresnel oscillations.  The smoothed version of despiked
     echo amplitudes is used to estimate the mean levels.  Some very large
     amplitude one pulse oscillations are removed by the despiking routine.
     The Fresnel pattern is getting close to aliasing at this point anyway
     so it does not matter much.
     The second run through does a linear interpolation between these points.
     The mean level profile looks better with a Pascal smoothing on this
     linear fit.  The final MeanAmp profile is an estimate of the base level
     of the meteor echo amplitudes.  That is an estimate of the profile in
     the absence of Fresnel effects.
```

```
      In the earlier versions of the Fresnel velocity determination I
      used the TriSmooth routine (with R = 3) to estimate the base level of
      the oscillations.  This was good for fast oscillations, near the end
      of the fresnel pattern but did not do so well near the start.
      23 July 1990 : Modified so as not to switch to SmoothRaw until at least
      three crossing points have been calculated in an effort to get past
      really 'long' initial oscillations.  Swings and Roundabouts; this now
      selects against small amplitude, 'high' velocity relatively short
      Fresnel patterns.  I prefer the slower velocity ones that this
      modification chooses.}

procedure SizeHalfCycles
    (var Residue:IntVector; Istart:integer; var CycleSize:HalfCycleSizes);
    {Searches through the residue values and finds the size of each half
     cycle.  This is used as a control number to fit a sine curve to each
     half cycle.  It is also used to decide where the end of the Fresnel
     oscillations occurs.}

procedure FindEndCycle
    (var Residue:IntVector; Istart:integer; var Iend:integer;
     var CycleSize:HalfCycleSizes; var EndCycle:shortint);
    {The routine checks through the half CycleSize data to find the end of
     the useful Fresnel oscillations.  It throws out ones with no Fresnel
     pattern and any patterns that do not last more than 3 full cycles (that
     is 0 - 5 half cycles).  In addition there must be at least 20 data points
     in the pattern to use the fitting routine.  EndCycle is returned as -4.
     Where these conditions are not satisfied the Fresnel oscillations can be
     lost into noise or become faster than the sampling frequency or just die
     out or result where the MeanAmp profile loses the center of the
     oscillations.  If the size of the last full cycle is greater than the
     size of the one before, then the Fresnel pattern is assumed to have
     ended with the half cycle before.  Where the maximum of a particular
     half oscillation is less than 4 A/D amplitudes from the MeanAmp value
     the oscillations are assumed to have become too small.  Additional
     checking against aliasing by the sampling rate is done within the
     GetFresnelDecel routine.}

procedure GetInvSinY
    (var CycleSize:HalfCycleSizes; Istart,EndCycle:integer;
     var Residue:IntVector; var InvSinY:RealVector);
    {Each half cycle of the Fresnel diffraction pattern is normalised
     by dividing by the maximum amplitude of the oscillation about the
     mean value (MeanAmp).  Each successive half cycle of the Fresnel
     diffraction pattern has an associated phase angle Pi larger than
     the one before.  The initial section from Imax to the first crossing
     point is not a good estimate of the initial Fresnel maximum.  This
     first (largest) half cycle is not used by the fitting routine anyway.
     The simple Fresnel theory with constant velocity, underdense ionisation
     and no allowance for decay of the trail is not so good for the early
     part of the trail either.  To ensure the angle is initialised correctly
     I do not evaluate InvSinY for the values from 0 to Pi.  Since cases
     occur where the Residue has two (or more) pulses with the same maximum
     value the routine only adds Pi on the FirstMax value of each half
     oscillation.}
```

```
procedure FitVelocity
    (FirstPoint,Iend:integer; var InvSinY:RealVector; Range:real;
     var Tzero,Velocity:real);
    {This code segment does a least squares fit to determine a Fresnel
     velocity on the data range specified.  The nominal reflection point
     Tzero is also calculated from the curve fit.  The "reflection point"
     corresponds to halfway up the rise to the initial maximum.
     1 August 1990, I have not made any further use of the Tzero
     information.  I forgot about it and very got back.}

procedure CheckVelocityStable
    (Istart:integer; var Iend:integer; var InvSinY:RealVector;
     var EndCycle:shortint; var CycleSize:HalfCycleSizes;
     Range:real; var VelSubDomain:RealVector);
    {It is necessary to check that the range of data points that are used to
     calculate the Fresnel velocity does not include any aliasing.  That is,
     the oscillations become too fast for the radar's pulse rate to detect
     all the peaks.  Noise fluctuations can also cause peaks to be missed.
     Where this happens the values of the GetInvSinY routine will be in
     error.  The only way I can think of to check for this is to calculate
     the velocity on successive subsets and look for a consistent rapid
     variation in the calculated velocity.  Graphing the result will also
     give an idea of the stability of the velocity calculated.
     25 July 1990.  This procedure now seems to be redundant.  Adding FirstMax
     to the GetInvSinY routine has eliminated the worst variations.  The new
     procedure GetFresnelDecel gives most of the information and is much
     quicker since it does not try to evaluate a velocity centered on every
     point.  This one is still useful as a display and diagnostics routine.}

procedure GetFresnelDecel
    (var Istart,Iend:integer; var EndCycle:shortint; var InvSinY:RealVector;
     var CycleSize:HalfCycleSizes; Range:real;
     var VelSubDomain:RealVector; var Beta,FresDecel:real);
    {Calculates the velocity on various sub ranges of the Fresnel pattern.
     The routine uses this to calculate a linear deceleration parameter for
     the echo profile.  The output vector VelSubDomain can be used as data
     for a graphics display of the Fresnel velocity data.  The routine fits
     the data from two complete oscillations for each velocity value.  This
     will weight the least squares fit of the deceleration parameter toward
     the more rapid later oscillations.  That is, weighted away from the first
     few oscillations where the simple theory is not so good.  There is no
     need to start and stop at integral half cycles; see CheckVelocityStable
     for example.
     30 July 1990: Adding code to stop the fit where the oscillations
     start to alias.  If the velocity jumps by 5 km.s-1 with the Fresnel
     fit on adjacent half cycles.  EndCycle and Iend are both changed
     accordingly.  In some cases the SketchMeanAmp routine misses the first
     one or few oscillations.  It generally causes the velocity sub domain
     to plunge substantially through the Fresnel pattern.  If FresDecel is
     greater than 120 km.s-2 it is rejected along with the associated Fresnel
     velocity.  I have seen some very good (consistent and believable)
     decelerations up around 100 km.s-2.  On this error the variable
     FresDecel is returned as 999.999 and EndCycle is changed to -5.}
```

```
procedure FresnelVelocity
    (var Smoothed,Site:SiteProfile; Range:real; var EndCycle:shortint;
     var VelSubDomain:RealVector; var Velocity:real);
    {Takes an echo profile from one receiver station and looks for a Fresnel
     diffraction pattern.  Where the routine identifies Fresnel oscillations
     it uses them to determine the velocity of the meteor producing the
     ionisation trail.  The mean amplitude level is removed from the raw
     signal to leave a residue.  A linearised least squares fit is done to
     estimate the velocity.  An estimate is made of the meteor's deceleration
     during the Fresnel oscillations.  The un-despiked amplitude profile is
     used.  The presence of heavy corona spikes will degrade the Fresnel code
     performance.}

procedure DrawDecelScales(var Buffer:ReducedData; VelLow:integer);
    {Draws a set scales to display the fresnel velocities calculated on
     subdomains of the available data.  Start the y scale at YLow, 10 km.s-1
     below the average Fresnel velocity.  The resultant plot shows the
     atmospheric deceleration of the meteor.  It also allows the reliability
     and stability of the calculated Fresnel velocity to be assessed.  This
     set of scales are specifically to combine VelSubDomain data from the
     three station amplitude data onto one graph.}

procedure PlotVelSubDomain
    (VelLow,SiteLag:integer; var SiteSub:RealVector; Symbol:SymbolType);
    {Plots the velocity sub domain for one site on the DrawDecelScales graph.}

procedure GetAtmosDecel
    (var AllSub,AllTime:RealVector; NumAll:integer; var Beta,AtmosDecel:real);
    {Does a least squares fit on the composite Fresnel velocity sub domain
     data.  This calculation is only called where there is data from two or
     more stations.  The actual least squares routine is very similar to the
     one in FitVelocity.  Where there is less than 8 data points the fit
     is abandoned and AtmosDecel set to 999.}

procedure AverageVelocities
    (V1,V2,V3:real; W1,W2,W3:integer; var NumFres:integer; var AvVel:real);
    {Sorts through the three velocities and finds the average value.  They
     must all agree to within 5km.s-1.  Rogue values are dropped.  The
     weights are just the number of half cycles used to determine the
     Fresnel velocities on each site.  The number of sites with velocities
     that agree is returned as NumFres.}

procedure GetFresnelVelocities
    (var ObsvBuff,SmoothBuff:Observation; ShowPlot:boolean;
     var Buffer:ReducedData);
    {The routine provides a shell for the Fresnel Velocity calculations.
     I am adding it to facilitate calculating the atmospheric deceleration
     from the Fresnel patterns on multiple station data.  The lags between
     each of the three profiles is calculated to allow the velocity sub
     domains to be compared on the same time scale.  FresnelVelocity is
     called for each of the receiver stations.  The average Fresnel
     velocity is noted and where Fresnel oscillations have been detected on
     more than one channel, an estimate of the atmospheric deceleration is
```

```
    made.  Where ShowPlot is passed as true the routine produces a graphics
    display of the deceleration data.}
```

B.7 OrbElems

```
unit OrbElems;                  {Conversion to Orbital Elements}
interface
uses   OrbDefns, GenUtil;

{This unit contains the procedures to convert the azimuth and zenith angles
 together with velocity of the observed meteor into a set of heliocentric
 orbital elements.}

{The following should be declared as constants in VAX/VMS Pascal.}
var    DiagnosticsOn : boolean;
       VRotation : real;              {Rotation velocity of observing station}
       sinA,cosA : real;              {Sin and Cos of the angle Alpha  }
       sinB,cosB : real;              {  and Beta.  Used as constants. }

type   ThreeVector = array[1..3] of real;
       TransMatrix = array[1..3,1..3] of real;
       DaysInYearArray = array[89..99] of integer;

const  BF_Latitude = -43.825;         {Lat. of station, 43deg 49min 30sec S}
       BF_Longitude = 11.512444;      {Long. of Birdlings  Flat  (hours) }
       D12 = 10.54;                   {Distance from Home to Nutt }
       Alpha = 45;                    {Angle to Nutt at Home from north }
       D13 =  8.18;                   {Distance from Home to Spit }
       Beta = 93;                     {Angle to Spit at Home from north }
       EE = 0.016693;                 {Earth's orbital eccentricity,  1989 }
       OneAU = 1.4959789e8;           {One AU in kms}
       G = 6.67204e-11;               {Universal gravitation constant}
       Msun = 5.9742e24 * 3.32958e5; {Mass of the sun, Mearth*(Msun/Mearth)}

var    Velocity : ThreeVector;        {Velocity vector of the meteor}
       A : TransMatrix;               {A generalised matrix for testing}
       DegToRad,RadToDeg : real;
           {Degree to radian and radian to degree conversion factors.  The
            compiler will not evaluate Pi/180 in a constant declaration.
            I have done it in the initialisation section.}


procedure GetOrbit
    (OutName:string; var Buffer:ReducedData; var Error:word);
    {This is the main procedure of this code unit.  It takes the reduced
     station data stored in the ReducedData type Buffer and calculates the
     elements of the orbit.  It uses my new technique of matrix
     multiplication to transform the observed velocity vector into the
     unperturbed (by the Earth) velocity vector and thence calculate the
     orbital elements.  The file called Reduc.Dat is used to store
     diagnostic output.}

procedure SetMatricies
```

```
    (var DaysInYear:DaysInYearArray; var A1,T,A2,A3,A4:TransMatrix);
    {This procedure initialises the coordinate transformation matrix arrays
     and the DaysInYear array.  Doing it explicity in this way allows for much
     easier transfer of the code to VAX/VMS Pascal.}

procedure Mul_AxV
    (A:TransMatrix; V:ThreeVector; var Y:ThreeVector);
    {Routine to multiply the 3x3 matrix A by the 3 vector V.
               --- j ------>
           |     11   12   13
           |     21   22   23      for i x j matrix A, A[i,j]
        i  |     31   32   33
           v
     The routine was tested by TMatMul.Pas with a number of matrices.}

procedure WriteOutMatrix
    (var OutFile:text; A:TransMatrix; V,Y:ThreeVector; Comment:string);
    {Writes out the matrix the vector it is operating on and the result.
     For output to the screen just use FileName:= ''  .}

procedure DisplayVector
    (var OutFile:text; VectorName:string; V:ThreeVector);
    {Writes out the vector V.  Output to screen, FileName:= ''  .}

procedure WriteOrbitElements
    (Buffer:ReducedData; OutName:string);
    {Writes out the orbital elements calculated by the GetOrbit routine and
     stored in the ReducedData buffer.  Screen output will be obtained by
     passing OutName as the null string ''    .}

function DayOfYear
    (Year,Month,Day:integer) : integer;
    {Calculates the day of the year.  The checking on the leap years
     should work for the years up to 2000.  The modified Julian date
     calculation the main orbit routine is based on the days since the
     start of 1989.}

function RMOD
    (Value,Denom : real) : real;
    {Returns the remainder after the Value/Denom.  Similar to mod but
     for real operands.}
```

B.8 GenUtil

```
unit GenUtil;              {General Utilities unit.}
interface
    uses Crt,DOS;

{Provides a set of commonly used routines.  These include string formating,
 reading the hardware clock and inverse trig functions.}

const  dchr : Array[0..9] of char = ('0','1','2','3','4','5','6','7','8','9');
       monthstr : Array[1..12] of string = ('Jan','Feb','March','April',
```

```
                'May','June','July','Aug','Sept','Oct','Nov','Dec');

function hexstr
    (x:word) : string;
    {converts a 16-bit integer to its hex representation as a string}

function intstr
    (x:longint) : string;
    {A function to produce the string representation of an integer. Written
     to avoid extra procedure calls}

function Datestr
    (Year,Month,Day : byte) : string;
    {Formats the date into a string for displaying observation information}

function Timestr
    (Hour,Min,Sec : byte) : string;
    {Formats the time into a string to display observation information.}

function CMOSDate
    (var Year,Month,Day : byte) : string;
    {Reads the hardware clock and returns the date as a string.}

function CMOSTime
    (var Hour,Min,Sec : byte) : string;
    {Reads the hardware clock and returns the NZST time.}

function InvSine
    (Ratio:real) : Real;
    {Returns the inverse sine of the ratio as an angle between -90 and 90.}

function InvCosine
    (Ratio:real) : Real;
    {Returns the inverse cosine of the ratio as a angle between -90 and 90.
     8 August 1990.  The routine is only returning values between 0 and 90.
     The previous form of the code should be o.k. for the elevation
     calculations;  For the elevation routines I explicitly place the angle
     between 0 and 180.  The problem developed when trying to use the
     routine with D-crit determinations.  A test bed is available in the
     program TestTrig.Pas.  The routine now produces resultant angles between
     0 and 180 degrees.}

function ArcSine(Ratio:real) : Real;
    {Returns the inverse sine as an angle between -Pi/2 and Pi/2.}

function ArcCosine(Ratio:real) : Real;
    {Returns the inverse cosine as an angle between 0 and Pi.}
```

B.9 OrbGraph

```
unit OrbGraph;                  {Meteor Orbit Graphics Utilities}
interface
```

```
uses
    OrbDefns, GenUtil, Graph, PostScript, Dos, Crt;

const MaxColor = 1;                        {For the HERC card on the meteor AT}

type  GrChannelType = record
                        name : string;           {of the channel}
                        initDI : word;           {first DI location}
                        x1,y1,x2,y2 : integer;   {viewport settings}
                      end;

var GrHome,GrSpit,GrNutt,
    GrTin,GrTos,Null : GrChannelType;    {5 A/D channels and a dummy}
    GrSmooth,GrPhase : GrChannelType;

procedure SetGraphics;
    {Detects and then initialises the graphics card.}

procedure AssociatePorts
    (var Top,Mid,Bot,TopR,MidR,BotR:GrChannelType);
    {Associates profile names with appropriate view port windows.}

procedure DrawProfilePort
    (var Channel:GrChannelType);
    {Puts the scale and labels on the profile display.}

procedure PlotProfile
    (var Channel:GrChannelType; var Site:SiteProfile; Noise:byte);
    {Plot the profile in the site record.}

procedure PlotPhase
    (var Site:SiteProfile; var Channel:GrChannelType; var Phase:Vector);
    {Plot the vector profile of a phase record.}

procedure PlotAngleVector
    (var Home:SiteProfile; Channel:GrChannelType; var Phase:PhaseVector);
    {Puts the scale and labels on the profile display.  Then plots the phase
     angle as calculated for each sweep.}

procedure DisplayObsv
    (var MeteorObsv:Observation);
    {Displays onto a graphics screen the contents of a meteor observation
     record.}

{Description of implementation procedures:
 InitChannels:  Puts data that is always associated with the graphics display
                of the 5 A/D channels into the ChannelType records. This
                routine is run on initialisation.}
```

B.10 OrbTests

```
unit OrbTests;                  {Meteor Orbit Radar Testing code.}
interface
```

```
    uses  OrbGraph, OrbDefns, GenUtil, Graph, Dos, Crt;

{A code unit for the procedures used while running the meteor orbit radar from
 the interactive Turbo environment.  For programs using this units MaxSweep
 should be set to TestMax = 400.  Several test and display routines provide
 several ways of viewing the data as a way of checking observing routines.}

{$F+}    {Force the far call mode to subroutines,
          especially for the interrupt service routine.}

{Andrew Taylor,   30 October  1989.}


procedure SaveDump
    (StartSweep:word);
    {Writes a MaxTestSweep sized AtoDMem dump to disk.  It interactively
     obtains a file name to save the memory dump to disk.  Typing return will
     prevent the data being saved.  The variable StartSweep allows the routine
     to make sure the sweep blocks that are saved with a meteor echo present
     are saved in the time order that they arrived in.}

procedure LoadDump;
    {Selects a data dump written by SaveDump and loads it into the AtoDMem.}

procedure AmpTrace;
    {Display the receiver amplitude traces of the Home, Nutt and Spit sites.
     The display starts at sweep one and scrolls through to MaxTestSweep.  It
     displays an average across 8 sweeps for each screen.}

procedure ProfileDump
    (var Channel:GrChannelType; var Site:SiteProfile);
    {Scans 400 sweep data dump time profiles and compares them with the
     profiles in the site vector.}

procedure HomePhaseDump
    (var Channel:GrChannelType; var Site:SiteProfile; var Tin,Tos:Vector);
    {Displays 400 sweeps of data from the phase channels and compares them
     with the amplitude profile at that range from the Home site.}

procedure AmplitudeDumps
    (var Channel:GrChannelType; var ObsvBuff:Observation);
    {Displays 400 sweeps of echo amplitude data from the RX channels and
     allows them to be compared with each other.}

{Implementation routines to allow graphics displays within the test routines.
 DrawDumpPorts :  Draws scales for displaying the 400 sweep AtoDMem dumps
                  collected for testing during development.
 DrawDumpVector :  Plots out the information stored in the Site record onto
                   a dump port.
 DrawDumpProfile :  Plots the profile given one range bin across the dump.
 AdjustRangeBin :  Changes the range bin given an input answer.}
```

B.11 OrbDists

```
unit OrbDists;                    {Orbit Distribution Plotting unit.}
interface
    uses  OrbDefns, Graph, PostScript, Crt, Dos;

{The unit provides a plotting package for use in looking at the reduced meteor
 orbit data.  This includes a number of arrays to tabulate the data and
 a series of plotting routines with which to display the data.  The unit
 assumes the postscript output file is controlled by the calling program. Any
 additional title information should be added by the calling program.  When
 the boolean AutoScale is false the plotting routines assume a scale
 appropriate for one day's data.  This will allow more direct comparison
 between data from each observing run.  With AutoScale true the routines
 scale the Y axis to fit all the cumulative counts.}

type  Array180 = array[0..180] of longint;
      Array180180 = array[-180..180] of longint;
      Array0_1 = array[0..210] of longint;          {0.005 A.U. steps}
      ArrayVel = array[-1..121] of longint;         {1 km.s-1 steps}
      Array360 = array[0..360] of longint;
      Array2_2 = array[-101..100] of longint;       {0.02 (A.U.)-1 steps}
      Array0_12 = array[0..121] of longint;         {0.1 A.U. steps}
      Array0_70 = array[0..70] of longint;          {6km range bins}
      Array256 = array[0..256] of longint;
      Array7_12 = array[70..120] of longint;         {1 km steps}
      Array20_20 = array[-41..42] of longint;
      Array40_80 = array[-41..82] of longint;

var   Num : longint;
      AutoScale : boolean;
      InclinArray : Array180;
      PeriHelArray : Array180180;
      PeriDistArray : Array0_1;
      SpeedHelioArray : ArrayVel;
      SpeedGeoArray : ArrayVel;
      FresnelArray : ArrayVel;
      EclipticLong : Array360;
      InvSemiArray : Array2_2;
      SpeedArray : ArrayVel;
      SemiMajorArray : Array0_12;
      ApDistArray : Array0_12;
      EccentricArray : Array0_12;
      HomeRangeArray,NuttRangeArray,SpitRangeArray : Array0_70;
      HomeAmpArray,NuttAmpArray,SpitAmpArray: Array256;
      ElevArray : Array360;
      PhaseArray : Array360;
      TinArray,TosArray : Array256;
      HomeAltArray,NuttAltArray,SpitAltArray : Array7_12;
      HomeDiffAltArray,NuttDiffAltArray,SpitDiffAltArray : Array7_12;
      LagsFresArray : Array20_20;
      AtmosDecelArray : Array40_80;
      FresDecelArray : Array40_80;
      LongAscArray : Array360;
```

```
      LongSunArray : Array360;

procedure ScaleYAxis
    (var YMax,Marks:integer);
    {This routine chooses the next 'good' value for YMax to ensure a
     pretty plot.  Any changes to the routine should make the appropriate
     adjustments within DrawNumberScale to keep the scaling correct.}

procedure DrawNumberScale
    (YMax,Marks:integer);
    {Draws the Number Y axis and scale onto the screen.  YMax should be
     a multiple of 10 to ensure correct scaling.}


procedure DrawSemiMajor
    (var SemiMajorArray:Array0_12; AutoScale:boolean);
    {Plots the section of the semi-major axis distribution from 0 to 12 A.U.
     This includes only reasonably short period elliptical orbits.  The data
     are plotted in 0.1 A.U. bins.  The number SemiMajorArray[0] counts those
     not included.}

procedure DrawLogSemiMajor
    (var SemiMajorArray:Array0_12);
    {A logarithmic version of the preceding procedure.}

procedure DrawInvSemiMajor
    (var InvSemiArray:Array2_2; AutoScale:boolean);
    {Plots the distribution of the inverse orbital semi-major axis.  This
     is plotted in 0.02 (A.U.)-1 steps.  The number InvSemiArray[-101]
     counts the number less than -2.  Effectively this means the meteoroid
     was on a very fast hyperbolic orbit.  The observational cutoff of 0.5AU
     for the semi-major axis always limits the inverse to less than 2.}

procedure DrawInclination
    (var InclinArray:Array180; AutoScale:boolean);
    {Draws a set of scales for the cumulative inclination plot.  Then plots
     the distribution.  InclinArray uses 1 degree bins for the count.}

procedure DrawArgumentPerihelion
    (var PerihelArray:Array180180; AutoScale:boolean);
    {Draws a set of scales for the cumulative argument of perihelion.  Then
     plots the distribution.  PerihelArray uses 1 degree bins for the count.
     The graph uses 2 degree bins to display the data.}

procedure DrawLongitude
    (var LongArray:Array360; LongName:string; AutoScale:boolean);
    {Draws the scales to display a longitude distribution.  Two uses
     immediately spring to mind; longitude of the ascending node and solar
     longitude at the time of observation.  The plot uses 2 degree bins.}

procedure DrawPerihelionDistance
    (var PeriDistArray:Array0_1; AutoScale:boolean);
    {The perihelion distance is plotted in steps of 0.005 A.U.}
```

```
procedure DrawHelioSpeed
    (var SpeedHelioArray:ArrayVel; AutoScale:boolean);
    {Plotting routine for the heliocentric speed of the meteoroid.  The data
     is grouped in 1 km.s-1 bins.  SpeedHelioArray[121] is the number of
     meteoroids with speeds greater than 120 km.s-1}

procedure DrawGeoSpeed
    (var SpeedGeoArray:ArrayVel; AutoScale:boolean);
    {Plotting routine for the corrected geocentric speed of the meteor.  The
     data is grouped in 1 km.s-1 bins.  SpeedGeoArray[121] is the number of
     meteors detected with speeds greater than 120 km.s-1}

procedure DrawFresnelSpeed
    (var FresnelArray:ArrayVel; AutoScale:boolean);
    {Plotting routine for the observed Fresnel velocities of the meteor.
     Where more than one station has produced a Fresnel velocity an average
     value AvFresVel is taken.  The ArrayVel[-1] entry counts how many
     observations obtained a Fresnel velocity on any station.}

procedure DrawObservedSpeed
    (var SpeedArray:ArrayVel; AutoScale:boolean);
    {Plot of the observed speed of the meteor.  This distribution is of the
     speed prior to any further processing.}

procedure DrawEclipticAngles
    (var EclipticLong:Array360; AutoScale:boolean);
    {Draws a diagram showing the ecliptic longitude of the meteoroids
     heliocentric velocity.  The corrected radiant points of the meteors
     are plotted in 4 degree bins.  I am not too sure about how well the
     scaling of the polar plot converts to printed postscript images.}

procedure DrawAphelionDist
    (var ApDistArray:Array0_12; AutoScale:boolean);
    {Plots the aphelion distance from 0 to 12 A.U.  This includes only
     reasonably short period elliptical orbits.  The data are plotted in
     0.1 A.U. bins.  The number ApDistArray[0] counts those not included.
     This procedure might be useful for looking for Jupiter or Saturn
     meteoroid families.  Do not find anything obvious.}

procedure DrawLogAphelionDist
    (var ApDistArray:Array0_12);
    {A logarithmic version of the preceding procedure.}

procedure DrawEccentricity
    (var EccentricArray:Array0_12; AutoScale:boolean);
    {Plot the eccentricity of the orbit.  The values range from 0 (circular)
     to 1.2 (hyperbolic) in steps of 0.01.  EccentricArray[121] contains the
     number of orbits more eccentric than 1.2.}

procedure DrawSiteRanges
    (var SiteRangeArray:Array0_70; SiteName:string; AutoScale:boolean);
    {Plots the range distribution for echoes from one receiver site.  The
     data is counted in 6km bins from 0 to 400km.  SiteRangeArray[70] is
     used to count those meteor trails with ranges greater than 400km.
```

```
     Observations at these longer ranges are now rejected so the number
     count will be zero.  Their velocities are too subject to timing
     uncertainties and phase angle ambiguities.}

procedure DrawSiteAmplitudes
    (var SiteAmpArray:Array256; SiteName:string; AutoScale:boolean);
    {Plots up a Site amplitude distribution.  Draws the amplitude values
     up the Y axis for a bit of variety.  The display uses steps of 4 A/D
     amplitude bins.}

procedure DrawElevations
    (var ElevArray:Array360; AutoScale:boolean);
    {Provides a polar plot of the number of meteors detected at the given
     elevation.  The data is plotted in 2 degree bins but plotted every
     0.5 degree to smooth the distribution.}

procedure DrawPhaseAngles
    (var PhaseArray:Array360; AutoScale:boolean);
    {Draws the cumulative distribution of the calculated phase angle.  These
     values will have the phase calibration constant already incorporated
     in them.}

procedure DrawTinTosAverages
    (var CountArray:Array256; PhaseName:string; AutoScale:boolean);
    {Plots the cumulative distributions for the average Tin and Tos values.}

procedure DrawSiteAltitudes
    (var SiteAltArray:Array7_12; SiteName:string; AutoScale:boolean);
    {Plots up a geometric altitude distribution for the reflection point
     associated with on receiver site.  Draws the altitude values up the
     Y axis like DrawSiteAmplitudes.  The display uses steps of 1 km
     altitude bins.  The routine adds on 5,10,25,50,75,90,95 percentile
     levels as well as the average altitude using the MarkDistLevels and
     MarkAverage.}

procedure DrawLagsFres
    (var LagsFresArray:Array20_20; AutoScale:boolean);
    {Draws the comparison histogram between the Fresnel velocities and the
     speed calculated from the three station time lags.  I am using 0.5 km.s-1
     steps and plotting the number against the difference of the two values.
     LagsFresArray[-41] and [41] are used to count those with a difference of
     more or less than  +-20 km.s-1.  LagsFresArray[42] counts the total
     number of comparisons made in the graph.}

procedure DrawDecelerations
    (var DecelArray:Array40_80; DecelName:string; AutoScale:boolean);
    {Plot of measured decelerations.  The routine is being written to plot
     both FresDecel (1 station) or AtmosDecel (2 or 3 station) Fresnel
     deceleration values.  Using 1 km.s-1 steps.  DecelArray[-41] and [81]
     are used to count those with a value of more or less than 80 km.s-2 or
     -40 km.s-2.  DecelArray[82] counts the total number of comparisons made.
     I now only calculate 2 or 3 station AtmosDecel values.}
```

B.12 OrbAstro

```
unit OrbAstro;                  {Astronomically useful procedures}
interface
    uses   OrbDists, OrbDefns, GenUtil, Graph ,PostScript;

{This unit contains procedures that use the reduced data (Orbit_**.) files
 to investigate features of astronomical interest.  Initially to do
 D-criteria calculations.  I am lifting the Drummond and Southworth-Hawkins
 criteria routines from a program I wrote in March 1987 !!  Originally it
 was used to evaluate the Adelaide meteor orbit data (3759 orbits only!).
 Time to use the functions on our own stuff.}

type  OrbitElements = record              {angles in radians}
                        PeriDist : real;  {Perihelion distance  (A.U.) }
                        Ecc : real;       {Eccentricity   }
                        Inclin : real;    {Inclination  (rad) }
                        ArgPeri : real;   {Argument of Perihelion (rad) }
                        LongNode : real;  {Longitude of ascending node, 1950}
                        EclipLat : real;  {Ecliptic latitude of perihelion}
                        EclipLong : real; {Ecliptic longitude of perihelion}
                      end;

const Nodes : array[36..56] of integer =
                 (1,2,3,0,0,4,5,6,7,8,0,9,10,11,12,0,0,0,13,14,14);
      Longs0700 : array[0..30] of real =
                 (0,36.65,37.65,38.62,41.54,42.51,43.47,44.43,45.42,47.35,
                  48.32,49.29,50.25,54.08,56.04,0,0,0,0,0,0,0,0,0,0,0,0,0,
                  0,0,0);

type  ArrayDCrit = array[0..101] of longint;
      ArrayADTLongs = array[-30..90] of longint;
      ArrayEtaLongs = array[0..30] of longint;
      ArrayEtaVel = array[350..480] of integer;

var   DegToRad : real;
      DDCritArray,SHCritArray : ArrayDCrit;
      EtaAquaridLongArray : ArrayEtaLongs;
      ADTLongArray : ArrayADTLongs;
      EtaHelioArray : ArrayEtaVel;

      etaAquaridOrbit,HalleyOrbit : OrbitElements;
      ADTetaAquaridOrbit : OrbitElements;


function DDcrit
    (var One,Two:OrbitElements; DDLevel:real) : real;
    {Calculates the Drummond D-criteria for the two sets of orbital elements
     given.  It represents the similarity between the two orbits.  The
     angles within the orbital elements must have previously been converted
     to radians.  Note that the routine does not explicity use the semi-major
     axis value for its calculation.  The calculation is only carried through
     where the value of DDCrit is less than DDLevel.  Where it is above DDCrit
     is returned as 999.9}
```

```
procedure DrawDcritDist
    (var DArray:ArrayDCrit; DCritName:string; AutoScale:boolean);
    {Plots the distribution of the D-criteria.  Either Drummond or
     Southworth-Hawkins variant.  Values with D > 1.00 are very
     dissimilar.  DArray[101] counts those.}

procedure DrawRADecScales;
    {Draws up a set of scales for the R.A. Dec plot.  The two associated
     plotting functions below put points onto these scales.}

function PlotRA(Xdata:real) : integer;
function PlotDec(Ydata:real) : integer;

procedure DrawEtaAquaridLongitudes
    (var LongArray:ArrayEtaLongs; AutoScale:boolean);
    {Draws a set of scales from 30 to 60 degrees solar longitude.  This
     provides a set of scales on which to plot the fine scale distribution
     of the eta Aquarid meteor shower.}

procedure DrawADTLongitudes
    (var LongArray:ArrayADTLongs; AutoScale:boolean);
    {Draws a set of scales from 300 to 90 degrees solar longitude.  This
     provides a set of scales on which to display the data collected as part
     of Andrew Taylor's PhD thesis (ADT).}

procedure DrawEtaRADecScales;
    {Draws up a set of scales for the enlarged R.A. Dec plot.  The axes show
     points in the ranges 320 < RA < 360 and -9 < dec < 3.  This provides an
     expanded scale for the eta Aquarid orbits.}

function PlotEtaRA(Xdata:real) : integer;
function PlotEtaDec(Ydata:real) : integer;

procedure DrawEtaHelioSpeed
    (var SpeedHelioArray:ArrayEtaVel; AutoScale:boolean);
    {Plotting routine for the eta Aquarid expanded distribution of the
     heliocentric speed of shower meteors. EtaHelioArray copes with anything
     between 35 and 48 km.s-1.  Routine plots everything between 36 and 48}

procedure DrawDecelVelScales;
    {Draws up a set of scales on which to plot the Fresnel velocity against
     Atmospheric deceleration.  The graph produces a scatter plot of points
     in the region  0 < v < 80 km.s-1  and  0 < decel < 60 km.s-2.  A set
     of high quality data (more than 12 half cycles) is available in the
     file C:\DATA\ DecelVel.Dat.}

procedure DrawLogDecelVelScales;
    {Draws a log-log plot of the same Velocity, Atmospheric deceleration
     data as the DrawDecelVelScales routine.}
```

B.13 OrbView

```
unit OrbView;          {To view orbit observation files on disk}
interface
    uses   OrbPost, OrbDefns, Unit_Mnu, Unit_Key,
           Graph, PostScript, Dos, Crt;

procedure SelectDataFile(DirName,FileMask:string; var FileName:string);
    {The routine looks at the data directory DirName and presents the files
     in a menu for selection.  If it contains no files with the specified
     mask then the routine terminates the program.  The menu uses
     POWER TOOLS plus routines.}

function SelectDataDir(var DirName:string; FileMask:string) : boolean;
    {The routine looks at the data directories in C:\DATA and presents them in
     a menu for selection.  The menu uses POWER TOOLS plus routines.  Returns
     false if the directory selected has no files of type FileMask.}

procedure DisplayDiskObsv
    (var ProcBuff:Observation; FileMask:string);
    {Looks at the observation records stored on disk.  Selecting the file to
     view is menu driven.  The last meteor observation viewed will be in the
     buffer ProcBuff when control is returned to the calling program.  Either
     by pressing  Q  to quit or  S  to select.  The routine uses the Blaise
     POWER TOOLS Plus menu routines.  If P is pressed a postscript copy of the
     display will be written to Output.PS }
```

B.14 NonPost

```
unit NonPost;         {Various routines without PostScript Graphics.}
interface
uses
    OrbDefns, GenUtil, Unit_Mnu, Unit_Key, Graph, Dos, Crt;

{To get hard copy graphics output calls to the Turbo graph unit are
 intercepted by the PostScript unit.  For many of the graphics routines
 the ghost procedures in this unit write postscript instructions out to
 a text file.  These files can then be transferred to the VAX and printed
 out on a postscript printer.  This slows down the PC's graphics display
 considerably.  I have assembled routines that do not use postscript to
 speed up display of the observation records where hard copy is required.
 Especially for SortOrbs (and LookDisk while observing).}

const MaxColor = 1;                        {For the HERC card on the meteor AT}

type  GrChannelType = record
                        name : string;           {of the channel}
                        initDI : word;           {first DI location}
                        x1,y1,x2,y2 : integer;   {viewport settings}
                      end;

var GrHome,GrSpit,GrNutt,
    GrTin,GrTos,Null : GrChannelType;     {5 A/D channels and a dummy}
```

```
    GrSmooth,GrPhase : GrChannelType;


procedure SetGraphics;
    {Detects and then initialises the graphics card.}

procedure AssociatePorts
    (var Top,Mid,Bot,TopR,MidR,BotR:GrChannelType);
    {Associates profile names with appropriate view port windows.}

procedure DrawProfilePort
    (var Channel:GrChannelType);
    {Puts the scale and labels on the profile display.}

procedure PlotProfile
    (var Channel:GrChannelType; var Site:SiteProfile; Noise:byte);
    {Plot the profile in the site record.}

procedure PlotPhase
    (var Site:SiteProfile; var Channel:GrChannelType; var Phase:Vector);
    {Plot the vector profile of a phase record.}

procedure PlotAngleVector
    (var Home:SiteProfile; Channel:GrChannelType; var Phase:PhaseVector);
    {Puts the scale and labels on the profile display.  Then plots the phase
     angle as calculated for each sweep.}

procedure DisplayObsv
    (var MeteorObsv:Observation);
    {Displays onto a graphics screen the contents of a meteor observation
     record.}

{Description of implementation procedures:
 InitChannels:  Puts data that is always associated with the graphics display
                of the 5 A/D channels into the ChannelType records. This
                routine is run on initialisation.}

procedure SelectDataFile(DirName,FileMask:string; var FileName:string);
    {The routine looks at the data directory DirName and presents the files
     in a menu for selection.  If it contains no files with the specified
     mask then the routine terminates the program.  The menu uses
     POWER TOOLS plus routines.}

function SelectDataDir(var DirName:string; FileMask:string) : boolean;
    {The routine looks at the data directories in C:\DATA and presents them in
     a menu for selection.  The menu uses POWER TOOLS plus routines.  Returns
     false if the directory selected has no files of type FileMask.}
```

B.15 PostScript

```
{*******************************************************}
{                                                       }
{       Turbo Pascal Version 5.0                        }
{       PostScript-Graph Unit                           }
{       Copyright (C) 1989 University of Canterbury;    }
{ parts Copyright (C) 1987,88 Borland International     }
{                                                       }
{*******************************************************}

unit PostScript;

interface

uses  GenUtil, Graph;


const PostscriptFilename : string[33] = 'OUTPUT.PS';
var PostscriptFile : text;

procedure PostOut(st : string);

{ *** detection, initialization and crt mode routines *** }

procedure SetGraphics;

procedure DeletePostScript;
procedure DeletePostScriptFile(PSFileName : string);
procedure OpenPostScript;
procedure NamePostScript(PSFileName:string);
procedure NewPostScriptPage;
procedure ClosePostScript;

procedure SetGraphMode(Mode : integer);
function GetGraphMode : integer;
procedure GraphDefaults;
procedure RestoreCrtMode;


{ *** Screen, viewport, page routines *** }
procedure ClearDevice;
procedure SetViewPort(x1, y1, x2, y2 : integer; Clip : boolean);
procedure ClearViewPort;

{ *** point-oriented routines *** }
procedure PutPixel(X, Y : integer; Pixel : word);

{ *** line-oriented routines *** }
procedure SetWriteMode(WriteMode : integer);
procedure LineTo(X, Y : integer);
procedure LineRel(Dx, Dy : integer);
procedure MoveTo(X, Y : integer);
procedure MoveRel(Dx, Dy : integer);
```

```
procedure Line(x1, y1, x2, y2 : integer);
procedure SetLineStyle(LineStyle : word;
                       Pattern   : word;
                       Thickness : word);

{ *** polygon, fills and figures *** }
procedure Rectangle(x1, y1, x2, y2 : integer);
procedure Bar(x1, y1, x2, y2 : integer);
procedure Bar3D(x1, y1, x2, y2 : integer; Depth : word; Top : boolean);
procedure DrawPoly(NumPoints : word; var PolyPoints);
procedure FillPoly(NumPoints : word; var PolyPoints);
procedure SetFillStyle(Pattern : word; Color : word);
procedure SetFillPattern(Pattern : FillPatternType; Color : word);
procedure FloodFill(X, Y : integer; Border : word);

{ *** arc, circle, and other curves *** }
procedure Arc(X, Y : integer; StAngle, EndAngle, Radius : word);
procedure Circle(X, Y : integer; Radius : word);
procedure Ellipse(X, Y : integer;
                  StAngle, EndAngle : word;
                  XRadius, YRadius  : word);
procedure FillEllipse(X, Y : integer;
                      XRadius, YRadius  : word);
procedure GetAspectRatio(var Xasp, Yasp : word);
procedure SetAspectRatio(Xasp, Yasp : word);
procedure PieSlice(X, Y : integer; StAngle, EndAngle, Radius : word);
procedure Sector(X, Y : Integer;
                 StAngle, EndAngle,
                 XRadius, YRadius : word);


{ *** color and palette routines *** }
procedure SetBkColor(ColorNum : word);
procedure SetColor(Color : word);
function GetBkColor : word;
function GetColor : word;
procedure SetAllPalette(var Palette);
procedure SetPalette(ColorNum : word; Color : shortint);
procedure GetPalette(var Palette : PaletteType);
function GetPaletteSize : integer;
procedure GetDefaultPalette(var Palette : PaletteType);
function GetMaxColor : word;
procedure SetRGBPalette(ColorNum, RedValue, GreenValue, BlueValue : integer);

{ *** bit-image routines *** }
function  ImageSize(x1, y1, x2, y2 : integer) : word;
procedure GetImage(x1, y1, x2, y2 : integer; var BitMap);
procedure PutImage(X, Y : integer; var BitMap; BitBlt : word);

{ *** text routines *** }
procedure GetTextSettings(var TextInfo : TextSettingsType);
procedure OutText(TextString : string);
procedure OutTextXY(X, Y : integer; TextString : string);
procedure SetTextJustify(Horiz, Vert : word);
```

```
procedure SetTextStyle(Font, Direction : word; CharSize : word);
procedure SetUserCharSize(MultX, DivX, MultY, DivY : word);


type  SymbolType = (Dot,Spot,OpenCirc,Plus,Cross,Box,FillBox);

procedure LabelYaxis(x,y:integer; TextString:string);
procedure OutTextCustom(x,y:integer; TextString:string);
procedure PlotPoint(X,Y:integer; Symbol:SymbolType; SymbolSize:integer);

const   Xscale : real = 0.711;
        Yscale : real = -1.1;
       Xoffset : real = 0.0;      {I'm tempted to just do it as a }
       Yoffset : real = 0.0;      {postscript translate in the initgraph}
       MyColor : word = 127;
     MyBkColor : word = 0;
  ScreenColors : word = 2;
WantPostscript : boolean = true;
```

Appendix C

Observation and Reduction Routines

This appendix includes the source code listings of those routines directly involved in the observation and reduction of meteor orbits. It contains the listings from the implementation sections of the relevant units. The two programs Orbserve.Pas and CalcOrbs.Pas which run the observing and data reduction are also included. The interface section of all the units used by the AMOR project are in Appendix B. The variables and type declarations of the definitions unit OrbDefns can be found there.

A number of routines have been made redundant during the development of the project. I have left these in the code units. They are not however included in the listings of this appendix. I have included the initialisation section of the code units where there is one. These initialisation sections are run at the start of the program that uses the unit.

Contents

C.1 Observe

```
program Observe;            {Meteor Orbit Observation Program.}
    uses  OrbLift, OrbRadar, OrbDefns, GenUtil, Crt, Dos;

{27 October  1989      Andrew Taylor.}

{This program operates the meteor orbit radar during an observing run.  It
 displays basic information about the run on the screen.  A graphical display
 of the observation records stored on disk can be obtained by running the
 program LookDisk.  The program clicks everytime it detects a new meteor echo
 on the Home site channel.  Data overrun conditions and any runtime errors
 cause the program to terminate.  The exit routines set off a high pitched
 alarm and save most of the current contents of the MetStore.  Bits of test
 code have been left as comments in the supporting units (especially the
 detection interrupt routine).}

begin
MaxSweep:= FieldMax;
SetAtoDMemory(AtoDMem,AtoDMemIRQ);
SetMetStore(MetStore);
{if CheckMemSizes then exit;           {A test option.}

HomeThreshold:= 30;
Answer:= ' ';
Port[DMACard]:= $28;
SetIntVec($0B,IntrDetector);
SetIntVec($60,IntrExtMemMove);
writeln('Press any key to save observations to disk');
writeln('Press S  to stop the run');

repeat                                    {Start of the main observing loop.}

GoToXY(1,4);
with DateStart do
    write(CMOSDate(Year,Month,Day),CMOSTime(Hour,Min,Sec):9);
writeln('System running':20);  writeln(' ':78);
GoToXY(1,8);  writeln('Last meteor echo detected');
SetIntVec($0B,IntrDetector);
Port[DMAcard]:= $F0;
Port[$21]:= $B5;           {disable IRQ#0 and enable IRQ#3}

while not StopRun do begin        {Start of the fill MetStore storage run.}
    if ReleaseNum <> LiftNum then begin
        if LiftNum = MaxDetect then LiftNum:= 1
            else inc(LiftNum);
        if DetectInfo[LiftNum].Number <= SaveNum then
            with DetectInfo[LiftNum] do        {A redetected persistent echo.}
                PersInfo[Number]:= PersInfo[Number] + PersTime
        else begin
            LiftMeteorObsv(LiftNum,ObsvBuff);   {A new echo detected.}
            Sound(600);
            with DetectInfo[LiftNum] do begin
                AtoDMem[DMAHoldFlag]^[0]:= 0;
```

```
            SaveNum:= Number;
          end;
          with ObsvBuff do begin
            Number:= SaveNum;  DetectionSite:= 1;
            GoToXY(32,8);
            writeln(Timestr(Hour,Min,Sec):12,Number:6);
          end;
          NeedToStore:= true;
          while NeedToStore do;              {Sycronise save to MetStore.}
          NoSound;
        end;
    end;

  {Run stopped if there is too much noise.}
    with ObsvBuff do if (N1 = 245) or (N2 = 245) or (N3 = 245) then Halt(50);
  {A transient noise spike can cause the whole trace to jump up.}
    i:= ReleaseNum - LiftNum;  if i < 0 then i:= i + MaxDetect;
    if i > 4 then Halt(51);
  {Stop the run if the observer wishes.}
    if KeyPressed then begin
        Answer:= ReadKey;  StopRun:= true;
    end;
  {Every 20 minutes ensure the observations collected are written to disk.}
    if PersCount >= 1200 then StopRun:= true;
  {If the extended memory MetStore is full then write it out to disk.}
    if SaveNum = MaxStore then StopRun:= true;

end;                                    {end of the fill MetStore storage run.}

Port[$21]:= $BC;           {enable timer (IRQ#0); disable IRQ#3 }
Port[DMACard]:= $F0;       {leave the TX trigger pulses going.}

GoToXY(1,5);  with DateEnd do writeln(CMOSDate(Year,Month,Day),
        CMOSTime(Hour,Min,Sec):9,'Data stopped':20);

GoToXY(1,16);  writeln('For the last observation storage run.');
writeln;  write('    data stopped   ');
with DateEnd do writeln(DateStr(Year,Month,Day),TimeStr(Hour,Min,Sec):9);
writeln('    run lasted ',PersCount,' seconds',' ':12);  writeln;
writeln('    meteors detected and saved to disk',SaveNum:5);
if PersCount = 0 then writeln('Run lasted for zero seconds')
    else writeln('Hourly Rate : ':39,(3600 div PersCount)*SaveNum:5);

WriteOutStore(SaveNum,DateStart,DateEnd,PersInfo);
CheckDiskFree;
ResetVars(AtoDMem,DetectCount,DetectNum,NewNum,ReleaseNum,LiftNum,SaveNum,
          PersInfo,PersCount,PersLow,SweepNum,AlarmCount,StopRun);

until UpCase(Answer) = 'S';                {end of the main observing loop.}

writeln('IRQCounter',IrqCounter:8);
end.
```

C.2 Run_Obsv

```
program  Run_Obsv;              {Parent program to run Observe}
    uses Genutil, Crt, Dos;

{Runs the program Observe under field conditions.  This parent program
 provides the error handling and ensures the observing code keeps running
 no matter what.  Messages are written out to the log file C:\ Obsv.Log.
 Run_Obsv terminates on getting a Halt(0) from the child process.  Noise
 and interference stops by Observe cause the program to pause for 10 seconds
 or 1 minute before trying again.  If Observe detects no meteors in 15
 minutes an alert alarm is sounded for 60 sec before trying again.  Once the
 free space on C: drive falls below 300KBytes an alert is sounded and the
 Run_Obsv program is terminated.}

{$M $4000,0,0}     {16K stack, no heap required or reserved.}

const LogName = 'C:\Obsv.Log';
      Delay_Noise = 10;     {delay ... 10 seconds }
      Delay_Spike = 60;     {delay ... 60 seconds }

type  TimeRec = record
                    Year,Month,Day : byte;
                    Hour,Min,Sec : byte;
                end;

var   Answer : char;
      ExitCodeNum : integer;
      Stop : boolean;
      LogTime : TimeRec;

procedure Alert;
    {Issues about a minute of warning noise to any observer present.  The
     procedure terminates early if a key is pressed.  The program continues
     running in either case.}
    var  i : integer;
         Answer : char;
begin
i:= 1;
while not KeyPressed and (i < 60) do begin
    Sound(4000); Delay(300); Sound(400); Delay(600); i:= i + 1;
end;
NoSound;  if KeyPressed then Answer:= ReadKey;
end;                                                    {end of Alert}


procedure WriteLog
    (var ExitCodeNum:integer; var Stop:boolean; var LogTime:TimeRec);
    {Writes out an entry in the log file.}
    var  LogFile : text;
         i : integer;
         Answer : char;
begin
assign(LogFile,LogName);
```

```
{$I-}  append(LogFile);  {$I+}
if IOResult <> 0 then rewrite(LogFile);

with LogTime do
    write(LogFile,CMOSDate(Year,Month,Day),CMOSTime(Hour,Min,Sec):9,' ':4);
case ExitCodeNum of
  -1 : {Entry code to start Run_Obsv.}
       begin
           for i:= 25 to 72 do write(LogFile,'*');  writeln(LogFile);
           writeln(LogFile,' ':24,'Run_Obsv started');
       end;
   0 : {Observer termination of Observe.  Run_Obsv stopped.}
       begin
           writeln(LogFile,'Run_Obsv terminated');  Stop:= true;
           for i:= 1 to 72 do write(LogFile,'*');  writeln(LogFile);
       end;
   1 : writeln(LogFile,'Run stopped to avoid overwriting AtoDMem data.');
   2 : writeln(LogFile,'Over run in DetectInfo    DetectNum = LiftNum');
   3 : writeln(LogFile,'DetectCount out of bounds.  DetectCount[i] too big');
   4 : begin
           writeln(LogFile,'5 minutes since the last meteor detection');
           Alert;
       end;
   5 : begin
           writeln(LogFile,'Less than 300KBytes on disk drive  C:');
           Alert;  Stop:= true;
       end;
  10 : writeln(LogFile,'An error in the extended memory transfer, intr $15.');
  20 : begin
           writeln(LogFile,'DosError of 2, 8, 10 or 11 with Exec procedure.');
           Alert;  Stop:= true;
       end;
  32 : begin  {for use with C:\EXE\Test.Exe press space bar.}
           Stop:= true;  writeln(LogFile,'Test.Exe stopped ******');
       end;
  50 : begin
           writeln(LogFile,'Too much noise Nx = 245.');
           for i := 1 to Delay_Noise do Delay(1000);
       end;
  51 : begin
           writeln(LogFile,'Transient noise spike more than 4 simultaneously.');
           for i := 1 to Delay_Spike do Delay(1000);
       end;
 101 : writeln(LogFile,'I/O error #101 : Disk write error (Full)');
 200 : writeln(LogFile,'Run time error #200 : Divide by zero');
 201 : writeln(LogFile,'Run time error #201 : Range check');
 203 : begin
           writeln(LogFile,'Run time error #203 : Heap overflow');
           Stop:= true;
       end;

  else begin
           writeln(LogFile,'ExitCodeNum',ExitCodeNum:4);
       end;
```

```
end;
close(LogFile);
end;                                          {end of WriteLog}

begin
ExitCodeNum:= -1;  Stop:= false;
WriteLog(ExitCodeNum,Stop,LogTime);

while not Stop do begin
    SwapVectors;
    Exec('C:\EXE\Observe.exe','');
    SwapVectors;
    if DosError <> 0 then ExitCodeNum:= 20
        else ExitCodeNum:= DosExitCode;

    WriteLog(ExitCodeNum,Stop,LogTime);
end;
end.
```

C.3 OrbRadar

```
procedure SetAtoDMemory
    (var AtoDMem:SweepPointers; var AtoDMemIRQ:SweepAddresses);
    {Aligns, allocates then calculates the addresses for the AtoDMem space.}
    var  i,j : integer;
         SpaceLeft : longint;
         StartSeg,DMAaddr : word;
         SweepNum : word;
begin
ClrScr;
SpaceLeft:= DiskFree(3);
if MaxSweep = FieldMax then  if SpaceLeft < 4000000 then begin
    str(SpaceLeft:10,Comment);  Sound(2000);
    writeln(Comment,' bytes free on Disk drive  C:');
    writeln('Press  R  to continue with run');
    Answer:= ReadKey;  NoSound;
    if UpCase(Answer) <> 'R' then Halt(0);
end;

writeln('Base of the heap',hexstr(seg(HeapPtr^)):8,hexstr(ofs(HeapPtr^)):6);

StartSeg:= Seg(HeapPtr^);                {Align heap for SweepBlocks of data}
StartSeg:= 16*( (StartSeg div 16) + 1 );
HeapPtr:= Ptr(StartSeg,0);
writeln('Base of AtoDMem ',hexstr(seg(HeapPtr^)):8,hexstr(ofs(HeapPtr^)):6);
write('Setting up the A to D memory space');
writeln('Size  =  ':14,MaxSweep,' sweeps');

for SweepNum:= 1 to MaxSweep do begin   {Allocate dynamic memory for DMA data}
    new(AtoDMem[SweepNum]);
    with AtoDMemIRQ[SweepNum] do begin
        StartSeg:= Seg(AtoDMem[SweepNum]^);
        DMAaddr:= StartSeg div 16;
```

```
        MSB:= DMAaddr mod 256;
        Page:= DMAaddr div 256;
    end;
   {zero memory locations for testing and to ensure any unused location
    can still be viewed in any diagnostic test program (parity set O.K.).}
    FillChar(AtoDMem[SweepNum]^,SizeOf(SweepBlock),#0);
end;
NoSound; ClrScr;
end;                                                    {end of SetAtoDMemory}


procedure ResetVars
    (AtoDMem:SweepPointers; var DetectCount:SweepBlock;
     var DetectNum,NewNum,ReleaseNum,LiftNum,SaveNum:byte;
     var PersInfo:PersistentTimes; var PersCount,PersLow:word;
     var SweepNum,AlarmCount:word; var StopRun,RunStopped:boolean);
    {Resets variables and DMAHold flag locations.}
    var  i : word;
begin
for i:= 1 to MaxSweep do
    AtoDMem[i]^[0]:= 0;                  {DMAHoldFlags}
for i:= 0 to 255 do
    DetectCount[i]:= 0;
DetectNum:= 0;  NewNum:= 0;  ReleaseNum:= 0;
LiftNum:= 0;  SaveNum:= 0;
PersCount:= 0;  PersLow:= 0;
FillChar(PersInfo,SizeOf(PersInfo),#0);
AlarmCount:= 0;
SweepNum:= 0;                {to start again at zero}
StopRun:= false;
end;                                                    {end of ResetVars}


procedure Detector;
    {Meteor orbit radar interrupt detection routine.}
interrupt;
{$R-}
    var    i,j,k : integer;
           LastSweep : SweepPtr;
           CountsSince : word;   {PersCount the last detection at that index.}
           SNum : byte;          {Last new detection at i,  DetectCount[i+4] }
           Regs : registers;
           EnoughTime : boolean;
           NumEchoes : byte;     {Number of simultaneous echoes for DMAHold.}
           DMAHoldLoc : integer; {To allow easy calculation (type integer) }
begin
Port[DMACard]:= $F1;     {take it high for timing.}
inc(IRQCounter);
EnoughTime:= true;
NumEchoes:= 0;

if SweepNum = MaxSweep then SweepNum:= 1
    else inc(SweepNum);
if SweepNum = 1 then LastSweep:= AtoDMem[MaxSweep]
```

```
    else LastSweep:= AtoDMem[SweepNum - 1];

if AtoDMem[SweepNum]^[0] = 255 then begin
    Comment:= 'Run stopped to avoid overwriting AtoDMem data.';
    Halt(1);
end;

with AtoDMemIRQ[SweepNum] do begin
    Port[$82]:= Page;       {Loading the page register in DMA controller}
    Port[$0C]:= $0;         {Reset byte pointer flip/flop}
    Port[$06]:= $1;         {LSB of base address register}
    Port[$06]:= MSB;        {MSB of base address register}
    Port[$07]:= $F9;        {Word count register: LSB}
    Port[$07]:= $00;        {                     MSB}
    Port[$0B]:= $07;        {Mode register to channel 3 demand write transfer}
    Port[$0A]:= $03;        {Clear mask bit register for channel 3}
end;
{Various pieces of test code
if (SweepNum = 0) then begin
    SetIntVec($0B,IntSave0B);
    Port[DMACard]:= $28;
    Finished:= true;
end;
if PersCount mod 10 = 8 then PersCount:= PersCount + 12;
}
{Check through the amplitude data for any meteors.  Home site.}
i:= 16;                                 {Start at range bin #18}
while i < 230 do                        {230 = Finish at range bin #60, Home}
    case DetectCount[i] of
       0   : begin
                 if LastSweep^[i] > HomeThreshold then
                     DetectCount[i]:= 1;
{                if i > 130 then DetectCount[i]:= 0;     {Test code.}
                 i:= i + 5;
             end;
     1..5  : begin
               if LastSweep^[i] > HomeThreshold then Inc(DetectCount[i])
                   else DetectCount[i]:= 0;
               i:= i + 5;
               end;
      255  : begin                      {Meteor data collection in process}
               i:= i + 5;
             end;
    7..205 : begin
               inc(DetectCount[i]);
               i:= i + 5;
               end;
       6   : begin                      {Positive meteor detection}
               EnoughTime:= false;
               if DetectNum = MaxDetect then DetectNum:= 1
                   else inc(DetectNum);
               if DetectNum = LiftNum then begin
                   Comment:= 'Over run in DetectInfo    DetectNum = LiftNum';
                   Halt(1);
```

```
        end;
        SNum:= DetectCount[i+4];
        if SNum = 0 then CountsSince:= 1000
            else  CountsSince:= PersCount
                    - word(DetectCount[i+2]) shl 8 - DetectCount[i+3];
        with DetectInfo[DetectNum] do begin
            if SweepNum = 1 then Detected:= MaxSweep
                else Detected:= SweepNum - 1;
            SweepOffset:= i;
            Threshold:= HomeThreshold;
            with Regs do begin
                AH:= $02;
                intr($1A,Regs);
                Hour:= 10*(CH shr 4) + (CH mod 16);
                Min:= 10*(CL shr 4) + (CL mod 16);
                Sec:= 10*(DH shr 4) + (DH mod 16);
            end;
            if CountsSince <= 2 then begin
                Number:= SNum;
                PersTime:= CountsSince;
                DMAHoldFlag:= 0;
            end else begin
                inc(NewNum);
                Number:= NewNum;
                SNum:= NewNum;
                PersTime:= 0;
                DMAHoldLoc:= integer(Detected) - 100 - NumEchoes;
                if (DMAHoldLoc > 0) then DMAHoldFlag:= DMAHoldLoc
                    else DMAHoldFlag:= DMAHoldLoc + MaxSweep;
                AtoDMem[DMAHoldFlag]^[0]:= 255;
                inc(NumEchoes);
            end;
        end;
        AlarmCount:= 0;
        DetectCount[ i ]:= 255;
        DetectCount[i+2]:= Hi(PersCount);
        DetectCount[i+3]:= Lo(PersCount);
        DetectCount[i+4]:= SNum;

        DetectCount[i-15]:= 255;        {Screen the active range bins}
        Move(DetectCount[i],DetectCount[i-10],5);
        Move(DetectCount[i],DetectCount[i-5],5);
        Move(DetectCount[i],DetectCount[i+5],5);
        Move(DetectCount[i],DetectCount[i+10],5);
        DetectCount[i+15]:= 255;

        DetectCount[i]:= 7;
        i:= i + 20;
        end;
206  : begin
        EnoughTime:= false;
        if ReleaseNum = MaxDetect then ReleaseNum:= 1
            else inc(ReleaseNum);
        j:= i - 15;
```

```
                for k:= 1 to 7 do begin          {Remove meteor active screen}
                    DetectCount[j]:= 0;
                    j:= j + 5;
                end;
                if i > 35 then begin             {Any active neighbours ?}
                    j:= i - 30;                     {Check closer range bins}
                    while j < (i - 15) do begin
                      if (DetectCount[j] > 5) and (DetectCount[j] < 255) then
                          for k:= 1 to 3 do begin
                              j:= j + 5;
                              DetectCount[j]:= 255;
                          end;
                      j:= j + 5;
                    end;
                end;
                if i < 220 then begin                {Check longer range bins}
                    j:= i + 30;
                    while j > (i + 15) do begin
                      if (DetectCount[j] > 5) and (DetectCount[j] < 255) then
                          for k:= 1 to 3 do begin
                              j:= j - 5;
                              DetectCount[j]:= 255;
                          end;
                      j:= j - 5;
                    end;
                end;
                i:= i + 20;
                end;
      else begin                  {An error condition}
            Comment:= 'DetectCount out of bounds     DetectCount['
                        + intstr(i) +'] = '+ intstr(DetectCount[i]);
            Halt(3);
           end;
    end;      {of the case options}

if PersLow = PersSweeps then begin
    inc(PersCount);
    PersLow:= 0;
    inc(AlarmCount);
    if AlarmCount >= 300 then begin
       {was 900 for 15 minutes rather than 5, 28/29 Apr 1991}
        Comment:= '5 minutes since the last meteor detection';
        Halt(4);
    end;
end else inc(PersLow);

Port[$20]:= $20;        {Enables interrupts to IRQ controller}
inline($FB);            {Enables further interrupts to CPU}

if NeedToStore and EnoughTime then begin
    ToExt:= true;
    intr($60,Regs);
    NeedToStore:= false;
end;
```

```
Port[DMACard]:= $F0;     {take it low for timing.}
{$R+}
end;                                              {End of the Detector routine}


procedure CheckDiskFree;
    {Check there is 300 KBytes free on the hard disk.}
    var  SpaceLeft : longint;
begin
SpaceLeft:= DiskFree(3);
if SpaceLeft < 308000 then begin
    str(SpaceLeft:10,Comment);
    Comment:= '    Disk drive  C:'+ Comment + ' bytes free';
    Halt(5);
end;
end;                                                    {end of CheckDiskFree}

procedure OrbitExit;
    {Exit procedure for error conditions in the OrbRadar.Pas unit.}
begin
ExitProc:= OrbitExitSave;

Port[$20]:= $20;              {reset interrupt controller.}
Port[$21]:= $BC;              {mask IRQ#3,  enable IRQ#0 for disk operations.}
SetIntVec($0B,IntSave0B);
if (ErrorAddr <> nil) or ((ExitCode > 0) and (ExitCode < 255)) then begin
    writeln;  write('PersCount  ',PersCount,'SweepNum  ':14,SweepNum);
    writeln('IRQCounter  ':20,IRQCounter);  writeln;
end;

if (ExitCode >= 1) and (ExitCode <= 4) then begin
    writeln('An error in the interrupt detection code.');
    writeln(Comment);
end;

if ExitCode = 5 then begin
    writeln('Insufficient disk space to continue.');
    writeln(Comment);
end;
end;                                                  {end of exit procedure}

{*************************************************************************}

begin                  {initialisation section}
NoSound;                    {Reset the speaker.}
ClrScr;
OrbitExitSave:= ExitProc;
ExitProc:= @OrbitExit;

GetIntVec($0B,IntSave0B);
IntrDetector:= @Detector;

IRQCounter:= 0;
StopRun:= false;
```

```
RunStopped:= false;
AlarmCount:= 0;
Comment:= '';

FillChar(DetectCount,SizeOf(DetectCount),#0);
SweepNum:= 0;  PersLow:= 0;  PersCount:= 0;  NewNum:= 0;
FillChar(DetectInfo,SizeOf(DetectInfo),#0);
DetectNum:= 0;  ReleaseNum:= 0;  LiftNum:= 0;

FillChar(ObsvBuff,SizeOf(ObsvBuff),#0);
FillChar(PersInfo,SizeOf(PersInfo),#0);
SaveNum:= 0;  ProcNum:= 0;
NeedToStore:= false;

HomeThreshold:= 30;  NuttThreshold:= 30;  SpitThreshold:= 30;

Finished:= false;
GotOne:= false;
end.
```

C.4 OrbLift

```
function FindProfile
    (StartSweep:word; OffsBase:word; var Offs:word; var Site:SiteProfile)
       : boolean;
    {Finds any associated meteor echoes on the other two channels.}
    var  Detected : boolean;
         Count,CountOffs : array[1..5] of word;
         i,j : word;
         SearchCount : byte;
begin
Detected:= false;
SearchCount:= 0;
Offs:= 0;
for j:= 1 to 5 do begin
    Count[j]:= 0;
    CountOffs[j]:= OffsBase + 5*j;
end;
i:= StartSweep;
with Site do begin
    while (not Detected) and (SearchCount < 200) do begin
        for j:= 1 to 5 do
            if AtoDMem[i]^[CountOffs[j]] < Threshold then
                Count[j]:= 0
              else
                if Count[j] = 6 then begin
                    Detected:= true;
                    DetectSweep:= i;
                    Offs:= CountOffs[j];
                end else
                    inc(Count[j]);
        if i = MaxSweep then i:= 1
            else inc(i);
```

```
        inc(SearchCount);
    end;
    RangeBin:= 15 + (Offs div 5);
    if SearchCount >= 200 then
        RangeBin:= 0;
    FindProfile:= Detected;
end;
end;                                                    {end of FindProfile}


procedure LiftProfile
    (var Site:SiteProfile; Offs:byte);
    {Picks an amplitude profile out of the AtoDMem space.}
    var  i : word;
         j,Offsminus,Offsplus : byte;
begin
with Site do begin
    if DetectSweep < 50 then
        i:= DetectSweep - 49 + MaxSweep
    else
        i:= DetectSweep - 49;
    IndexOne:= i;
    Offsminus:= Offs - 5;
    Offsplus:= Offs + 5;
    for j:= 1 to 250 do begin
        Profile[j]:= (AtoDMem[i]^[Offsminus] + (AtoDMem[i]^[Offs] shl 1)
                     + AtoDMem[i]^[Offsplus]) shr 2;
        if i = MaxSweep then i:= 1
            else inc(i);
    end;
end;
end;                                                    {end of LiftProfile}

procedure RangeScanOffset
    (var Site:SiteProfile; Offs:byte; var ROffs:byte);
    {Three point fit to improve Home site range estimation.}
    var  i : word;
         j,k,Offsminus,Offsplus : byte;
         Amp : byte;
         Ranges : array[1..16] of real;
         Sum : real;
begin
with Site do begin
    Offsminus:= Offs - 5;
    Offsplus:= Offs + 5;
    if DetectSweep <= 4 then i:= DetectSweep - 4 + MaxSweep
        else  i:= DetectSweep - 4;
    j:= 1;  k:= 46;
    repeat
        Amp:= Profile[k];
        Ranges[j]:= (15.5*AtoDMem[i]^[Offsminus] + 1.3*AtoDMem[i]^[Offs]
                              - 18.2*AtoDMem[i]^[Offsplus]) / Amp;
        inc(j);  inc(k);
        if i = MaxSweep then i:= 1
```

```
            else inc(i);
    until (j > 12) or (Amp > 160);
end;

Sum:= 0;
for i:= 1 to j - 1 do Sum:= Sum + Ranges[i];
ROffs:= round(100 + Sum / j);
end;                                                    {end of RangeScanOffset}


procedure LiftPhase
    (var Site:SiteProfile; var Phase:Vector; Offs:byte);
    {Picks out the phase channel amplitudes associated with the echo.}
    var  i : word;
         j,Offsminus,Offsplus : byte;
begin
i:= Site.IndexOne;
for j:= 1 to 250 do begin
    Phase[j]:= AtoDMem[i]^[Offs];
    if i = MaxSweep then
        i:= 1
    else
        inc(i);
end;
end;                                                        {end of LiftPhase}

procedure LiftNoise
    (var Site:SiteProfile; Offs:byte);
    {Takes an amplitude sample from the channel when no echo was detected.}
    var  i : word;
         j,Offsminus,Offsplus : byte;
begin
with Site do begin
    i:= IndexOne;
    Offsminus:= Offs - 5;
    Offsplus:= Offs + 5;
    for j:= 1 to 250 do begin
        Profile[j]:= (AtoDMem[i]^[Offsminus] + (AtoDMem[i]^[Offs] shl 1)
                     + AtoDMem[i]^[Offsplus]) shr 2;
        if i = MaxSweep then i:= 1
            else inc(i);
    end;
end;
end;                                                        {end of LiftNoise}

procedure NoiseSample
    (Threshold,initOffs:byte; StartSweep:word; var Noise:byte);
    {Measures the noise level for several sweeps around the detection sweep.}
    var  i : word;
         j,Offs : byte;
         Sum,Count : word;
begin
i:= StartSweep;
Sum:= 0; Count:= 0;
```

```
for j:= 1 to 8 do begin
    Offs:= initOffs;
    while Offs < 250 do begin
        if AtoDMem[i]^[Offs] < Threshold then begin
            inc(Count);
            Sum:= Sum + AtoDMem[i]^[Offs];
        end;
        Offs:= Offs + 5;
    end;
    if i = MaxSweep then i:= 1
        else  inc(i);
end;
if Count < 170 then Noise:= 250
    else  Noise:= Sum div Count;
end;                                                    {end of NoiseSample}


procedure LiftMeteorObsv
    (LiftNum:byte; var ObsvBuff:Observation);
    {Lifts a meteor observation out of the AtoDMem into ObsvBuff.}
    var    StartLook : word;
           Offs,NuttOffs,SpitOffs,TinOffs,TosOffs : word;
begin
ObsvBuff.DetectionSite:= 1;
with DetectInfo[LiftNum] do begin
    StartLook:= DMAHoldFlag;
    Offs:= SweepOffset;
    ObsvBuff.Year:= 0;
    ObsvBuff.Month:= 0;
    ObsvBuff.Day:= 0;
    ObsvBuff.Hour:= Hour;
    ObsvBuff.Min:= Min;
    ObsvBuff.Sec:= Sec;
end;
with ObsvBuff do begin
    with Home do begin
        DetectSweep:= DetectInfo[LiftNum].Detected;
        RangeBin:= 15 + Offs div 5;
        Threshold:= HomeThreshold;
    end;
    LiftProfile(Home,Offs);
    RangeScanOffset(Home,Offs,RangeOffset);
    with Nutt do begin
        NoiseSample(NuttThreshold,12,Home.DetectSweep,N2);
        Threshold:= N2 + 10;
        if FindProfile(StartLook,Offs+1,NuttOffs,Nutt) then
            LiftProfile(Nutt,NuttOffs)
        else begin
            NuttOffs:= Offs + 16;
            DetectSweep:= Home.IndexOne;
            IndexOne:= DetectSweep;
            RangeBin:= 15 + NuttOffs div 5;
            LiftNoise(Nutt,NuttOffs);
        end;
```

```
    end;
    with Spit do begin
        NoiseSample(SpitThreshold,13,Home.DetectSweep,N3);
        Threshold:= N3 + 10;
        if FindProfile(StartLook,Offs+2,SpitOffs,Spit) then
            LiftProfile(Spit,SpitOffs)
        else begin
            SpitOffs:= Offs + 17;
            DetectSweep:= Home.IndexOne;
            IndexOne:= DetectSweep;
            RangeBin:= 15 + SpitOffs div 5;
            LiftNoise(Spit,SpitOffs);
        end;
    end;
    TinOffs:= Offs + 3;
    LiftPhase(Home,Tin,TinOffs);
    TosOffs:= Offs + 4;
    LiftPhase(Home,Tos,TosOffs);
    NoiseSample(HomeThreshold,11,Home.DetectSweep,N1);
end;
end;                                                    {end of LiftMeteorObsv}


procedure SetMetStore
    (var MetStore:StoreAddresses);
    {Sets up the extended memory MetStore addresses.}
    var  Temp : longint;
         i : word;
begin
Temp:= $100000;                          {Start at the 1MByte boundary.}
for i:= 1 to MaxStore do begin
    with MetStore[i] do begin
        HiByte:= byte(Temp div 65536);
        LoWord:= word(Temp mod 65536);
    end;
    Temp:= Temp + BuffSize;
end;
end;                                                      {end of SetMetStore}


procedure InitGDT
    (var GDT:GlobalDescTable);
    {Sets up the features of the Global Descriptor Table common to both read
     and write operations with extended memory.}
begin
FillChar(GDT,SizeOf(GDT),#0);
with GDT do begin
    with Source do begin
        SegLimit:= BuffSize;
        DataAccess:= $93;
    end;
    with Target do begin
        SegLimit:= BuffSize;
        DataAccess:= $93;
```

```
    end;
end;
end;                                                    {end of InitGDT}

procedure SetGDTbones
    (var GDTread,GDTwrite:GlobalDescTable);
    var    Temp : longint;
begin
InitGDT(GDTwrite);
Temp:= 16*longint(Seg(ObsvBuff)) + longint(Ofs(ObsvBuff));
with GDTwrite.Source do begin
    HiByte:= byte(Temp div 65536);
    LoWord:= word(Temp mod 65536);
end;

InitGDT(GDTread);
Temp:= 16*longint(Seg(ObsvBuff)) + longint(Ofs(ObsvBuff));
with GDTread.Target do begin
    HiByte:= byte(Temp div 65536);
    LoWord:= word(Temp mod 65536);
end;
end;                                                 {end of SetGDTbones}


procedure ExtMemMove;
    {Moves the contents of ObsvBuff to and from the extended memory store.}
interrupt;
    var  GDT : GlobalDescTable;
         Reg : registers;
begin
inline($FB);                              {Enable further interrupts}
if (ToExt) then begin                     {Move to extended memory}
    GDT:= GDTwrite;
    with GDT.Target do begin
        HiByte := MetStore[SaveNum].HiByte;
        LoWord := MetStore[SaveNum].LoWord;
    end;
    NeedToStore:= false;
  end else begin                          {Copy from extended memory}
    GDT:= GDTread;
    with GDT.Source do begin
        HiByte := MetStore[ProcNum].HiByte;
        LoWord := MetStore[ProcNum].LoWord;
    end;
end;

with Reg do begin                         {Call BIOS $15, subfunction $87}
    AX:= $8700;
    CX:= BuffSize shr 1;
    ES:= Seg(GDT);
    SI:= Ofs(GDT);
    Intr($15,Reg);
    if ((Flags and FCarry) <> 0) then begin
        Comment:= 'An error in the extended memory transfer, intr $15.';
```

```
      Halt(3);
    end;
end;
end;                                                    {end of ExtMemMove}


procedure WriteOutStore
    (SaveNum:byte; DateStart,DateEnd:DateRecord;
     var PersInfo:PersistentTimes);
    {Writes the observations stored in the extended memory out to disk.}
    const   dchr : Array[0..9] of char
                          = ('0','1','2','3','4','5','6','7','8','9');
    var     Directory,FileName : string;
            HourFile : byte;                    {Hour associated with FileName}
            OutFile : file of Observation;
begin
HourFile:= 30;
for ProcNum:= 1 to SaveNum do begin
    ToExt:= false;
    intr($60,Regs);
    if PersInfo[ProcNum] > 254 then ObsvBuff.PersTime:= 255
        else ObsvBuff.PersTime:= PersInfo[ProcNum];
    with ObsvBuff do begin
        if (DateStart.Day = DateEnd.Day) or (Hour > 12) then begin
            Year:= DateStart.Year;
            Month:= DateStart.Month;
            Day:= DateStart.Day;
        end else begin
            Year:= DateEnd.Year;
            Month:= DateEnd.Month;
            Day:= DateEnd.Day;
        end;
        if Hour <> HourFile then begin
            if ProcNum > 1 then Close(OutFile);
            Directory:= dchr[Year div 10] + dchr[Year mod 10] +
                        dchr[Month div 10] + dchr[Month mod 10] +
                        dchr[Day div 10] + dchr[Day mod 10];
            Directory:= 'C:\DATA\' + Directory;
            FileName:= 'NZST_' + dchr[Hour div 10] + dchr[Hour mod 10];

     {$I-}  MkDir(Directory);  {$I+}              {Ensure directory is there}
            if IOResult <> 0 then;                {Directory already exists}
            Assign(OutFile,Directory + '\' + FileName);
     {$I-}  Reset(OutFile);  {$I+}
            if IOResult <> 0 then Rewrite(OutFile);
            Seek(OutFile,FileSize(OutFile));       {Find the end of the file}
            HourFile:= Hour;                       {A new data file is open}
        end;
    end;
    write(OutFile,ObsvBuff);                    {Write out the buffer}
end;
if Hourfile < 30 then Close(OutFile);
end;                                                  {end of WriteOutStore}
```

```
procedure LiftExit;
    {Exit procedure to ensure that most of the data in the MetStore is saved
     should a runtime error terminate the program.}
    var  DummyStr : string;
begin
ExitProc:= LiftExitSave;

Port[$20]:= $20;              {reset interrupt controller.}
Port[$21]:= $BC;              {mask IRQ#3,  enable IRQ#0 for disk operations.}
writeln;

if ExitCode = 10 then begin   {An error in the extended memory transfer.}
    writeln(Comment);
    for i:= 1 to 5 do begin
        Sound(3000);  Delay(400);  Sound(4000);  Delay(400);
    end;
    NoSound;
end;

if (ErrorAddr <> nil) or ((ExitCode > 0) and (ExitCode < 255)) then begin
    with DateEnd do
        write(CMOSDate(Year,Month,Day),CMOSTime(Hour,Min,Sec):9);
    writeln('Run terminated on  ERROR':26, ExitCode);
    {was getting this from OrbLift, so see what it is... 5 May 1991}
    if SaveNum > 1 then begin
        writeln(SaveNum-1,' observations saved to disk.');
        WriteOutStore(SaveNum - 1,DateStart,DateEnd,PersInfo);
    end;
end;
SetIntVec($60,IntSave60);
NoSound;
end;                                                {end of exit procedure}

{*************************************************************************}

begin                                              {initialisation section}
LiftExitSave:= ExitProc;
ExitProc:= @LiftExit;
SetGDTbones(GDTread,GDTwrite);

GetIntVec($60,IntSave60);
IntrExtMemMove:= @ExtMemMove;
end.
```

C.5 CalcOrbs

```
program CalcOrbs;          {Reduce observations and get orbital elements}
    uses  OrbElems, OrbFres, OrbLags, OrbReduc, OrbDefns,
          GenUtil, Crt, Dos;

{The program reads in data from <name>.Orb files.  It then reduces the
 observation records and calculates the orbital elements of the original
```

```
 meteoroid.  Information is put out to Orbit_<hr>.ReD as a ReducedData record.
 14 May 1990.  The program has been modified to search for all the *.Orb data
 files within the named directory and process the lot.  Output to Orbit_<hr>.
 13 October 1990.  Modified to use the new max rising slope method for
 determining the timelags between the stations.  ReducSweep has been defined
 as a constant in OrbReduc; should reject a few less now.  New deceleration
 formula for the atmospheric correction is introduced.  Determining the
 decay constant now uses the differentiated profile.  Also using the
 despiked observation profiles for the Fresnel calculations.  This is a
 definite 11th hour mod.  The program also contains a drafting option to
 check its decisions about what to include.  Where Fresnel velocities are
 calculated on more than one channel an additional check is made to ensure
 that they agree to within 5km.s-1 before a atmospheric deceleration is
 estimated.}

label   999,   {Problem with time lag or elevation data.  Cannot get orbits.}
        888,   {Data reduction routines could not get good aeronomy data.}
        777;   {User option to terminate the program with  S  or  Q .}

var   DirName : string;                          {Location of input data file}
      HourName : string;                           {File name less the extension}
      InName,OutName : string;
      InData : file of Observation;
      RecNum : longint;                            {Record number in the file}
      Report : text;                               {Report file from ScreenEcho}
      ReportName : string;
      Answer : char;
      LagSum,LagDiff,RelativeDiff : real;
      Ambig : boolean;                             {If still an ambiguous phase}
      PhaseStr,RangeStr : string;                  {Strings for error messages}
      RemoteElev : real;                           {Elevation, remote echo point}
      RemoteRange : real;                          {Range from Home to echo point}
      TrailLength : real;                          {Home to point to RX (km)}

      OKFile,BadFile : file of Observation;
      Buffer : ReducedData;
      OrbData : file of ReducedData;
      ShowFile : text;
      i : integer;

      SFile : SearchRec;
      SearchName : string;


begin
TestGraphicsOn:= false;
DiagnosticsOn:= false;
DirName:= 'D:\900503';
DirName:= 'D:\901004';
DirName:= 'C:\DATA';

assign(Report,DirName +'\Report');
rewrite(Report);
```

```
SearchName:= DirName + '\NZST_*.tst';
FindFirst(SearchName,AnyFile,SFile);
while DosError = 0 do begin                          {Start loop in Directory}

ClrScr;
{Delete the current contents of the diagnostics file.}
assign(ShowFile,'Reduc.Dat');
rewrite(ShowFile);
close(ShowFile);

assign(ShowFile,'Show.tmp');
rewrite(ShowFile);

InName:= DirName +'\'+ SFile.Name;
writeln('Processing through file :  ',InName);  writeln;
with Buffer do
    writeln(Report,CMOSDate(Year,Month,Day),CMOSTime(Hour,Min,Sec):10);
writeln(Report,'Report on orbit rejections from :  ',InName);
assign(InData,InName);
reset(InData);  read(InData,ObsvBuff);
with ObsvBuff do ReadPhaseCalib(Year,Month,Day,PhaseCalib);
reset(InData);
seek(InData,0);

OutName:= 'Orbit_'+ SFile.Name[6] + SFile.Name[7];
OutName:= DirName +'\'+ OutName;
writeln(Report,'Reduced Data in ReducedData file :  ',OutName);
writeln(Report);
writeln('Data to ReducedData file :  ',OutName);
assign(OrbData,OutName);
rewrite(OrbData);
Window(1,5,80,24);

{
assign(OKFile,DirName + '\NZST_' + SFile.Name[6] + SFile.Name[7] + '.OK');
rewrite(OKFile);
assign(BadFile,DirName + '\NZST_' + SFile.Name[6] + SFile.Name[7] + '.Bad');
rewrite(BadFile);
}

{
read(InData,ObsvBuff);
while not eof(Indata) and (ObsvBuff.Min < 30) do
    read(InData,ObsvBuff);
}

while not eof(InData) do begin
{
for i:= 1 to 20 do begin
}
    read(InData,ObsvBuff);
    with ObsvBuff do
        writeln(Datestr(Year,Month,Day):12,Timestr(Hour,Min,Sec):10,Number:6);
```

```
    Error:= 0;
    FillChar(Buffer,SizeOf(Buffer),#0);
    TransferObsvData(ObsvBuff,Buffer);

    with ObsvBuff do begin
        DeSpikeProfile(Home.Profile,N1);
        DeSpikeProfile(Nutt.Profile,N2);
        DeSpikeProfile(Spit.Profile,N3);
    end;


    TriSmooth(ObsvBuff.Home,SmoothBuff.Home,3);
    with Buffer.Home do
        FindMax(SmoothBuff.Home,Imax,Max,250);

    TriSmooth(ObsvBuff.Nutt,SmoothBuff.Nutt,3);
    with Buffer.Nutt do
        FindMax(SmoothBuff.Nutt,Imax,Max,250);

    TriSmooth(ObsvBuff.Spit,SmoothBuff.Spit,3);
    with Buffer.Spit do
        FindMax(SmoothBuff.Spit,Imax,Max,250);

{Use the MaxRiseSlopeLags routine to find the timelags.}
    MaxRiseSlopeLags(SmoothBuff,Buffer,
            Buffer.Lag12,Buffer.Lag13,Buffer.Lag23,Error);
    if Error > 0 then begin
        Comment:= 'Ambiguous maximum rising slope point';
        goto 999;
    end;
    with Buffer do if ( (Lag12 > 0) and (Lag13 < 0) and (Lag23 > 0) )
        or ( (Lag12 < 0) and (Lag13 > 0) and (Lag23 < 0) ) then begin
        Error:= 10;
        Comment:= 'Illegal combination of time lags';
        goto 999;
    end;

{This section finds the elevation angle and range for the ionisation trail.}
    with Buffer do begin
        MeanTinTos(Home.Max,SmoothBuff.Home,ObsvBuff.Tin,AvTin,SdevTin);
        MeanTinTos(Home.Max,SmoothBuff.Home,ObsvBuff.Tos,AvTos,SdevTos);
        if (AvTin > 256) or (AvTos > 256) then begin
            Comment:= 'Not enough values for Tin and Tos';
            Error:= 1; Goto 999;
        end;

        Get_Phase(Year,Month,Day,Hour,PhaseCalib,
                AvTin,SdevTin,AvTos,SdevTos,Phase,PhasDiff);
        if (PhasDiff >= 5) then begin
            str(PhasDiff:5:1,Comment);
            Comment:= 'Large uncertainty in Phase' + Comment;
            Error:= 15; Goto 999;
        end;
```

```
        Get_Ranges(ObsvBuff,Buffer);

        Get_Elevation(Phase,Elev);
        Get_Altitude(Elev,Home.Range,Home.Altitude);
        ResolvePhase(Phase,Buffer.Lag23,Elev,Home.Range,Home.Altitude,Ambig);
        if (Phase > 310) and (Phase < 350) and (Home.Range < 220) then begin
            Comment:= 'Range ambiguity,  Range > 400 km or not?';
            Error:= 14;  goto 999;
        end;
        if Home.Range > 400 then begin
            str(Home.Range:4,RangeStr);
            Comment:= 'Has an ambiguous range.  Range '+ RangeStr;
            Error:= 13;  goto 999;
        end;
        if Ambig then begin
            str(Home.Range:4,RangeStr);  str(Phase:4:0,PhaseStr);
            Comment:= 'Has an ambiguous phase.  Range '+ RangeStr
                      + '  Phase '+ PhaseStr;
            Error:= 6;  goto 999;
        end;
        if (Elev < 4) or (Elev > 176) then begin
            str(Elev:4:0,Comment);
            Comment:= 'Has an unlikely elevation.  Elev'+ Comment;
            Error:= 11;  goto 999;
        end;
    end;

{This section tries to calculate the velocity of the meteor based on any
 Fresnel oscillations that may be present in the amplitude record.}

    GetFresnelVelocities(ObsvBuff,SmoothBuff,false,Buffer);


{The next section reduces the aeronomy related data.  As such it is not
 necessary for calculating meteoroid orbits.}

    with Buffer.Home do begin
        GetDecay(ObsvBuff.Home,SmoothBuff.Home,Imax,Noise,Decay,Error);
        if (Decay <= 0) or (Decay > 250) then begin
            Error:= 7;  str(Decay:4,Comment);
            Comment:= 'Problem fitting Home site decay.  Decay '+ Comment;
        end else begin
            DiffHeight:= Decay / 314.65;
            DiffHeight:= 79.0 - 7.24*ln(DiffHeight);
        end;
    end;

    with Buffer.Nutt do begin
        GetDecay(ObsvBuff.Nutt,SmoothBuff.Nutt,Imax,Noise,Decay,Error);
        if (Decay <= 0) or (Decay > 250) then begin
            Error:= 7;  str(Decay:4,Comment);
            Comment:= 'Problem fitting Nutt site decay.  Decay '+ Comment;
        end else begin
            DiffHeight:= Decay / 314.65;
```

```
            DiffHeight:= 79.0 - 7.24*ln(DiffHeight);
        end;
    end;

    with Buffer.Spit do begin
        GetDecay(ObsvBuff.Spit,SmoothBuff.Spit,Imax,Noise,Decay,Error);
        if (Decay <= 0) or (Decay > 250) then begin
            Error:= 7;  str(Decay:4,Comment);
            Comment:= 'Problem fitting Spit site decay.  Decay '+ Comment;
        end else begin
            DiffHeight:= Decay / 314.65;
            DiffHeight:= 79.0 - 7.24*ln(DiffHeight);
        end;
    end;


  888: {Error handling for problems with the quality of the aeronomy data.}
      if Error > 0 then begin
      { 7 : Could not fit the decay constant to the amplitude data.
                Decay =   0, Not enough points to calculate decay constant.
                Decay = 252, Decay goes out of range 0 to 250 pulses.
                Decay = 253, Profile shows less than 1/e decay in amplitude.
                Decay = 254, Fit out of range.
                Decay = 255, A significant rising edge after maximum
                         amplitude.  The decay profile is affected by wind.
       18 : Could not establish the decay constant.
       }
          Error:= 0;
      end;


{Calculate the orbital elements for the meteoroid.}
    GetOrbit('',Buffer,Error);       {'Reduc.Dat'  is also an option.}
    case Error of
       21 : Comment:= 'Observed velocity less than escape velocity';
       22 : Comment:= 'Azimuth not consistent with time lags';
      end;
    if Error > 0 then goto 999;

{Calculate the altitude for the reflection points of the two remote sites.}
    with Buffer do begin
        Nutt.Altitude:= Vz * Lag12 / Prf;
        Nutt.Altitude:= Home.Altitude - Nutt.Altitude;

        Spit.Altitude:= Vz * Lag13 / Prf;
        Spit.Altitude:= Home.Altitude - Spit.Altitude;
    end;

{
    with ObsvBuff do
        writeln(ShowFile,Datestr(Year,Month,Day):12,Timestr(Hour,Min,Sec):10,
                         Number:6,Buffer.Elev:12:0,Buffer.Speed:8:1);
}
    write(OrbData,Buffer);
```

```
{
    write(OKFile,ObsvBuff);
}

  999: {Error handling for data reduction problems which affect the data
        necessary to calculate orbits.  Error numbers greater than 20
        originate in the GetOrbit procedures.}
       if Error > 0 then begin
           with ObsvBuff do
           writeln(Report,Datestr(Year,Month,Day):12,
                          Timestr(Hour,Min,Sec):10,Number:6);
           writeln(Report,' ':32,Comment);
           writeln(Report,' ':32,'No data recorded for this observation');
           writeln(' ':32,Comment);
           writeln(' ':32,'No data recorded for this observation');
       { 1 : There were not enough sweeps above 20 at the Home site
                 to give a good average for Tin or Tos.
         2 : Within the time lags routine the parabolic fit went out of the
                 expected range.
         3 : No sensible maximum in the CCF of the cross correlation routine.
         4 : The CCF is too flat to give a good cross correlation peak.
         5 : The amplitude of the three time lags are too inconsistent with
                 each other.
         6 : No sensible combination of the range and phase angle is
                 possible.  The phase provides an ambiguous elevation angle
                 that puts the meteor trail above TooHigh or below TooLow.
         9 : One of the profiles has its maximum late in the recorded profile.
                 Its small size precludes using just the rising edge to get
                 time lags.
        10 : No meteor could give this combination of the three time lags.
        11 : Elevation angle implied by the phase is too low (Elev < 4 deg).
        12 : The time lags are all too short.  Any velocities calculated would
                 probably be meaningless.  Meteor windblown or with a very
                 low elevation angle.
        13 : The observation is appearing in sweep of the next radar pulse.
                 This means the time lags are measured in a plane almost
                 perpendicular to the meteor velocity;  Too inaccurate.
        14 : The geometry cannot distinguish between ranges of 150 or 550 km.
                 This will reject about 5 to 8 percent.  Half of them would
                 be rejected for range > 400 km anyway.
        15 : Uncertainty in the phase angle is greater than 10 degrees.
        17 : Error in MaxRiseSlopeLags.  Either fit to max slope out of range
                 or the rising has ambiguous points of maximum slope.
        19 : Failed CheckOut : No detection on both remote site profiles.
                 26 Oct, 1990 : Have moved this check to ScrnObsv.
        21 : Observed velocity less than escape velocity.
        22 : Azimuth not consistent with the sectors implied by time lags.
        }
           Error:= 0;
{
           write(BadFile,ObsvBuff);
}
       end;
```

```
    {test write out of the reduced data.}
    WriteReducedData(Buffer,'reduc.dat');

    if KeyPressed then begin
        Answer:= ReadKey;
        if (UpCase(Answer) = 'Q') or (UpCase(Answer) = 'S') then begin
            writeln(Report,'Reduction run  TERMINATED BY USER');
            writeln;  writeln;
            writeln('Reduction run  TERMINATED BY USER');
            goto 777;
        end;
    end;
end;                                              {end of eof(InData) loop}

RecNum:= FilePos(InData);
writeln(Report,RecNum,'  Observation records in file');
writeln;  writeln(RecNum,'  Observation records in file');

Window(1,1,80,25);  GoToXY(1,1);

close(InData);
close(OrbData);
close(ShowFile);
FindNext(SFile);
end;                                               {end of Directory loop}

777: {An exit routine for a user terminated run.}
if (UpCase(Answer) = 'S') or (UpCase(Answer) = 'Q') then begin
    close(InData);
    close(OrbData);
    close(ShowFile);
{
    close(OKFile);
    close(BadFile);
}
end;

close(Report);
Window(1,1,80,25);  GoToXY(1,25);
end.
```

C.6 OrbReduc

```
procedure TransferObsvData
    (var ObsvBuff:Observation; var Buffer:ReducedData);
    {Puts the data from the observation record into a ReducedData record.}
begin
Buffer.Year:= ObsvBuff.Year;
Buffer.Month:= ObsvBuff.Month;
Buffer.Day:= ObsvBuff.Day;
Buffer.Hour:= ObsvBuff.Hour;
Buffer.Min:= ObsvBuff.Min;
Buffer.Sec:= ObsvBuff.Sec;
```

```
Buffer.Number:= ObsvBuff.Number;
Buffer.PersTime:= ObsvBuff.PersTime;

Buffer.Home.Noise:= ObsvBuff.N1;
Buffer.Nutt.Noise:= ObsvBuff.N2;
Buffer.Spit.Noise:= ObsvBuff.N3;
end;                                                {end of TransferObsvData}


procedure TriSmooth
    (var Site,SmoothSite:SiteProfile; R:byte);
    {Triangular smoothing function to a range of R elements.}
    var  WI : array[1..12] of real;      {Weighting function}
                {Dimensioned > R to stop dynamic dimensioning.}
         i,j,k : byte;
         Dummy : real;
begin
for k:= 1 to R do
    WI[k]:= (R-k+1)/(R+1);
SmoothSite:= Site;
with Site do
  for i:= R+1 to 250-R do begin
    Dummy:= Profile[i];
    for k:= 1 to R do begin
        j:= i - k;
           Dummy:= Dummy + WI[k]*Profile[j];
        j:= i + k;
           Dummy:= Dummy + WI[k]*Profile[j];
    end;
    SmoothSite.Profile[i]:= byte(Round(Dummy/(R+1)));
  end;
end;                                                {end of TriSmooth}


procedure FindMax
    (var Site:SiteProfile; var Imax,Max:byte; LimitIndex:byte);
    {Finds magnitude and location of the maximum amplitude in the profile.}
    var  i : byte;
begin
Max:= 0;
with Site do for i:= 10 to LimitIndex do
    if Profile[i] > Max then begin
        Max:= Profile[i];
        Imax:= i;
    end;
end;                                                {end of FindMax}


procedure DifferentiateProfile
    (var Amp:SiteProfile; Start,Finish:integer; var AmpDash:IntVector);
    {Differentiates the amplitude profile in the range Start to Finish.}
    const  w = 5;
    var    i : integer;
begin
```

```
if Start < 16 then Start:= 16;
if Finish > 235 then Finish:= 235;
for i:= Start to Finish do with Amp do begin
    AmpDash[i]:= - Profile[i - 3*w] + 9*Profile[i - 2*w] - 45*Profile[i - w]
                 + 45*Profile[i + w] - 9*Profile[i + 2*w] + Profile[i + 3*w];
end;
end;                                          {end of DifferentiateProfile}


procedure GetDecay
    (var Site,Smoothed:SiteProfile; Imax,Noise:integer;
     var Decay:byte; var Error:word);
    {Estimate an exponetial decay constant for the profile.}
    label  888;
    var   Istart,Iend : integer;
          i,j,y : integer;
          x : longint;
          AmpDash : IntVector;
          MaxDash,MinDash : integer;
          MaxDecay : integer;              {index of max decay, MinDash.}
          CutPoint : real;
          AreaRise : real;
          s : string;
          NumPos : integer;

          SumX,SumY : real;
          SumSqrX,SumXY : real;
          Num : word;
          MeanX,MeanY : real;
          Sxx,Sxy,Syy : real;
          Beta,Tau : real;
begin
DifferentiateProfile(Smoothed,Imax,235,AmpDash);

MinDash:= 0;  MaxDecay:= 0;
for i:= Imax to 235 do if AmpDash[i] < MinDash then begin
    MinDash:= AmpDash[i];
    MaxDecay:= i;
end;
if MaxDecay = 0 then begin
    Error:= 18;  Decay:= 254;  goto 888;
end;
Istart:= MaxDecay - 5;

if TestGraphicsOn then begin
    DrawDiffProfile(Site,Imax,235,AmpDash);
    MaxDash:= - MinDash;
    DifferentiateProfile(Smoothed,16,Imax,AmpDash);
    for i:= 16 to Imax do if abs(AmpDash[i]/MaxDash) < 1.078 then
      PlotPoint(2*i,round(64*(1 - AmpDash[i]/MaxDash)),Dot,1)
    else
      PlotPoint(2*i,round(64*(1 - 1.078*AmpDash[i]/abs(AmpDash[i]))),Plus,2);

    Line(0,48,500,48);                    {0.25 of the most negative}
```

```
    Line(2*MaxDecay,128,2*MaxDecay,48);  {At most negative point}
    Line(2*Istart,256,2*Istart,48);
end;

{Check there are no significant rising edges.}
j:= MaxDecay;  CutPoint:= abs(0.25*MaxDash);  AreaRise:= 0;
while (AmpDash[j] < CutPoint) and (AreaRise < 2400) and (j < 234) do begin
    if AmpDash[j] > 0 then AreaRise:= AreaRise + AmpDash[j]
        else AreaRise:= 0;
    j:= j + 1;
end;
if (AmpDash[j] >= CutPoint) or (AreaRise >= 2400) then begin
    if TestGraphicsOn then begin
        OutTextXY(500,150,'Error : Second rising edge');
{
        OutTextXY(440,162,'NumPos  '+ intstr(NumPos));
}
        str(AreaRise:6:0,s);
        OutTextXY(440,162,'AreaRise  '+ s);
    end;
    Error:= 18;  Decay:= 255;  goto 888;
end;

{Find the first min point after max decay slope.}
j:= MaxDecay;
while (AmpDash[j] < 0) and (j < 235) do j:= j + 1;
if TestGraphicsOn then begin
    PlotPoint(2*j,64,Plus,6);  Line(2*j,64,2*j,160);
end;

{First noise after the min point for end of decay fit.}
while (Smoothed.Profile[j] > Noise + 5) and (j < 250) do j:= j + 1;
Iend:= j;
if TestGraphicsOn then Line(2*Iend,268,2*Iend,160);

SumX:= 0 ;  SumY:= 0;  SumXY:= 0;  SumSqrX:= 0; Num:= 0;
{If the unsmoothed profile has a zero value pretend it is actually 1.}
for i:= Istart to Iend do begin
    SumX:= SumX + i;  x:= i;
    SumSqrX:= SumSqrX + sqr(x);
    if Site.Profile[i] = 0 then begin
        SumY:= SumY + ln(1);
        SumXY:= SumXY + i*ln(1);
    end else begin
        SumY:= SumY + ln(Site.Profile[i]);
        SumXY:= SumXY + i*ln(Site.Profile[i]);
    end;
    Num:= Num + 1;
end;

if Num > 1 then begin
    MeanX:= SumX / Num;
    MeanY:= SumY / Num;
    Sxx:= SumSqrX - sqr(SumX) / Num;
```

```
    Sxy:= SumXY - SumX*SumY/Num;

    if Sxy = 0 then Sxy:= 1.0e-10;
    Tau:= - Sxx/Sxy;
    if (Tau < 0) or (Tau >= 250) then begin
        Decay:= 252;  goto 888;
    end else Decay:= round(Tau);
    Beta:= MeanY + MeanX/Decay;

    if TestGraphicsOn then begin
        if abs(Decay) > 4 then DrawDecay(Istart,Iend,exp(Beta),Decay,Noise);
        SetTextJustify(LeftText,CenterText);
        OutTextXY(400,160,'Num '+ intstr(Num));
        OutTextXY(400,150,'Decay  '+ intstr(round(Decay)));
    end;
end else Decay:= 0;

if Num < 15 then Decay:= 0;
{Not enough points to give a good estimate on the decay.}

if (Decay > Num + 10) then Decay:= 253;
{Insist that the data profile record a drop of 1/e to claim the decay as
 valid.  The profile sample is only 250 sweeps long which will mean this
 routine will select only reasonably rapid decays.}

if Decay < 5 then Decay:= 253;
{The routine could not sensibly get a decay this small.}

if TestGraphicsOn then if (Decay > 250) or (Decay <= 0) then
    OutTextXY(400,180,'Rejected');

888:
end;                                                {end of GetDecay}


procedure MeanTinTos
    (Max:byte; var SmoothHome:SiteProfile; var VT:Vector;
     var Average,StdDev:real);
    {To find the mean and std. dev. of the Tin or Tos vectors.}
    var  i,m,y,StartAmp : word;
         Sum,Sum2 : real;
begin
Sum:= 0.00;  Sum2:= 0.00;  M:= 0;  i:= 40;

if Max < 24 then StartAmp:= 16
    else StartAmp:= 7*Max div 10;
while (SmoothHome.Profile[i] < StartAmp) and (i <= 120) do
    i:= i + 1;

while (m < 24) and (i <= 120) do begin
    if SmoothHome.Profile[i] > 16 then begin
        m:= m + 1;
        Sum:= Sum + VT[i];
        Sum2:= Sum2 + sqr(VT[i] + 0.0);
```

```
        if TestGraphicsOn then PlotPoint(i,85 - VT[i] div 3,Spot,1);
    end;
    i:= i + 1;
end;
if (m < 7) then begin
    if (m < 4) or ( (m >= 4) and (StdDev < 20) ) then begin
        Average:= 999;
        StdDev:= 0;
    end;
end else begin
    Average:= Sum / m;
    StdDev:= sqrt((Sum2 - Average*Sum)/(m-1));
    StdDev:= StdDev / sqrt(m);              {std error of the mean.}
end;
end;                        {of MeanTinTos}

procedure ReadPhaseCalib
    (Year,Month,Day:byte; var PhaseCalib:HourValues);
    {An Initialisation routine for the PhaseCalib array.}
    var  CalFile : text;
         DateStr : string;
         DummyStr : array[1..6] of char;
         i,Calib : integer;
         StartHour,EndHour : byte;
begin
DummyStr:= '******';
for i:= 0 to 23 do PhaseCalib[i]:= +20;
DateStr:= char(Year div 10 + ord('0')) + char(Year mod 10 + ord('0'))
           + char(Month div 10 + ord('0')) + char(Month mod 10 + ord('0'))
           + char(Day div 10 + ord('0')) + char(Day mod 10 + ord('0'));

assign(CalFile,'PhaseCal.Dat');
reset(CalFile);
while not eof(CalFile) and (DateStr <> DummyStr) do begin
    readln(CalFile);
    for i:= 1 to 6 do  if not eoln(CalFile) then
        read(CalFile,DummyStr[i]);
end;
while not eoln(CalFile) do begin
    read(CalFile,StartHour,EndHour,Calib);
    for i:= StartHour to EndHour do PhaseCalib[i]:= Calib;
end;

close(CalFile);
end;                                                {end of ReadPhaseCalib}

procedure FitPhase_1983
    (AvTin,AvTos:real; var Phas:real);
    {Empirical fit of phase angle from the Tin and Tos values.}
begin
    IF (AvTin > 0) AND (AvTos > 0) THEN
IF (avTIN > TopTin) OR (avTIN < BotTin) THEN BEGIN
    IF avTIN > TopTin THEN Phas:= (376 - avTOS)/0.98;
    IF avTIN < BotTin THEN Phas:= (avTOS - 30)/1.10;
```

```
  END ELSE BEGIN
    IF avTOS > 128 THEN Phas:= (avTIN + 58)/1.17;
    IF avTOS < 128 THEN Phas:= (533 - avTIN)/1.19;
END;
if Phas < 0 then Phas:= Phas + 360;
if Phas >= 360 then Phas:= Phas - 360;
end;                        {end of FitPhase_1983}

procedure Get_Phase
    (Year,Month,Day,Hour:byte; var PhaseCalib:HourValues;
     AvTin,SdevTin,AvTos,SdevTos:real; var Phas,PhasDiff:real);
    {Finds the phase difference between the two home receivers.}
    var  ErrorPhas : real;

       {This was -40 for the May 1989 runs.  The phase seems to have been
        inverted (see Get_Phase commments) and the phase calibration needs
        changing.  It puzzles me why this is the case;  10 Jan 1990.}
       {It was -60 for the days data 900503 and 900504. Home site aerials
        were connected to the wrong RX boxes.  At some stage in the last
        year I shortened the North coax line into the RX.}


begin
FitPhase_1983(AvTin,AvTos,Phas);

if (AvTin > TopTin) or (AvTin < BotTin) then
    if AvTos > 128 then FitPhase_1983(AvTin,AvTos - SdevTos,ErrorPhas)
            else  FitPhase_1983(AvTin,AvTos + SdevTos,ErrorPhas)
else
    if AvTin > 128 then FitPhase_1983(AvTin - SdevTin,AvTos,ErrorPhas)
            else  FitPhase_1983(AvTin + SdevTin,AvTos,ErrorPhas);
PhasDiff:= abs(ErrorPhas - Phas);
if PhasDiff > 340 then PhasDiff:= 360 - PhasDiff;

{Since the last runs in May 1989 I seem to have swapped the sense of the
 inputs into the phase comparison circuits.  It therefore needs to be
 inverted.}
Phas:= 360 - Phas;


{Reduction of observations made prior to June 1989 have the old sense.}
if (Year = 89) and (Month < 6) then Phas:= 360 - Phas;

{The data collected on 900503 and 900504 had the Home aerials plugged into the
 wrong RX boxes.  The sense of the phase difference is inverted.}
if (Year = 90) and (Month = 5) then  if (Day = 3) or (Day = 4) then
    Phas:= 360 - Phas;


Phas:= Phas + PhaseCalib[Hour];
if Phas < 0 then Phas:= Phas + 360;
if Phas >= 360 then Phas:= Phas - 360;

end;                        {end of Get_Phase}
```

```
procedure Get_Ranges
    (var ObsvBuff:Observation; var Buffer:ReducedData);
    {Finds the range to the reflection points on the meteor trail).}
begin
Buffer.Home.Range:= 6*ObsvBuff.Home.RangeBin - HomeDelay;
if ObsvBuff.RangeOffset > 1 then with Buffer.Home do
    Range:= Range - round((ObsvBuff.RangeOffset - 100)/6.67);
Buffer.Nutt.Range:= 6*ObsvBuff.Nutt.RangeBin - NuttDelay;
Buffer.Spit.Range:= 6*ObsvBuff.Spit.RangeBin - SpitDelay;
end;                        {end of Get_Ranges}

procedure Get_Elevation
    (Phas:real; var Elev:real);
    {Calculates the elevation angle to the Home site reflection point.}
begin
if Phas < 320 then begin
    Phas:= Phas*Pi/180;
    Elev:= (Phas/(2*Pi) + 4.0) / AerialSpacing;
end else begin
    Phas:= Phas*Pi/180;
    Elev:= (Phas/(2*Pi) + 3.0) / AerialSpacing;
end;
Elev:= InvCosine(Elev);
end;                                                {end of Get_Elevation}

procedure Get_Altitude
    (Elev,Range:real; var Altitude:real);
    {Establishes the altitude. May resolve phase and range ambiguities.}
    var  AltFlat : real;
begin
Elev:= Elev*Pi/180;              {Convert to radians}
AltFlat:= Range*sin(Elev);      {Flat Earth approximation}
Altitude:= 2*Range*EarthsRadius*sin(Elev) + sqr(Range);
Altitude:= Altitude / (2*EarthsRadius + AltFlat);
end;                                                 {end of Get_Altitude}


procedure ResolvePhase
    (Phas,Lag23:real; var Elev:real; var Range:integer; var Altitude:real;
     var Ambig:boolean);
    {Resolves phase ambiguities with time lags and range phase distribution.}
begin
Ambig:= true;

if Lag23 <= 0 then begin                {Meteor trail to the south}
    if (Altitude > TooLow) and (Altitude < TooHigh) then begin
       {These values would seem to be unambiguous.}
        Ambig:= false;
        if TestGraphicsOn then PlotASymbol(Spot,Range,Phas,1);
    end;
    if Ambig and (Altitude < TooLow) and (Phas > 50) and (Phas < 320)
        and (Range > 110) and (Range <= 160) then begin
       {Elevation angles 36 to around 50.}
```

```
        if TestGraphicsOn then  PlotASymbol(FillBox,Range,Phas,2);
        Phas:= Phas;
        Elev:= (Phas/(360) + 3.0) / AerialSpacing;
        Elev:= InvCosine(Elev);
        Get_Altitude(Elev,Range,Altitude);
        if (Altitude > TooLow) and (Altitude < TooHigh) then
            Ambig:= false;
    end;
    if Ambig and ( (Phas > 300) or (Phas < 20) ) then begin
       {Elevation between 13 and 4 degrees.  Range > 396.}
        if TestGraphicsOn then  PlotASymbol(Cross,Range,Phas,2);
        Range:= Range + 396;
        Elev:= (Phas/(360) + 4.0) / AerialSpacing;
        if (Elev > 1) and (Elev < 1.1) then begin
            Elev:= 1;
            Ambig:= true;
        end else
            Elev:= InvCosine(Elev);
        Get_Altitude(Elev,Range,Altitude);
        if (Altitude < TooHigh) and (Altitude > 10) then  Ambig:= false;
       {NOTE the altitude is probably in considerable doubt anyway.}
    end;
    if TestGraphicsOn then if Ambig then
        PlotASymbol(Cross,Range,Phas,4);

end else begin                          {Meteor echo from behind (north)}
    if Phas > 40 then begin
       {Assume that it occurs between elevation 144 and 167 initially.}
        Phas:= 360 - Phas;
        Elev:= (Phas/(360) + 4.0) / AerialSpacing;
        Elev:= InvCosine(Elev);
        Get_Altitude(Elev,Range,Altitude);
        Elev:= 180 - Elev;
    end else begin
       {Assume that it occurs between elevation 140 and 144 initially.}
        Phas:= 360 - Phas;
        Elev:= (Phas/(360) + 3.0) / AerialSpacing;
        Elev:= InvCosine(Elev);
        Get_Altitude(Elev,Range,Altitude);
        Elev:= 180 - Elev;
    end;
{NOTE: Phase is now 360 - Phase as though the trail were to the south.}
    if (Altitude > TooLow) and (Altitude < TooHigh) then begin
        Ambig:= false;
        if TestGraphicsOn then  PlotASymbol(OpenCirc,Range,360 - Phas,2);
    end;
    if Ambig and (Altitude < TooLow) and (360-Phas > 50) and (360-Phas < 320)
        and (Range > 110) and (Range <= 160) then begin
       {Elevation angles 144 to around 130.}
        if TestGraphicsOn then  PlotASymbol(Box,Range,360 - Phas,2);
        Elev:= (Phas/(360) + 3.0) / AerialSpacing;
        Elev:= InvCosine(Elev);
        Elev:= 180 - Elev;
        Get_Altitude(Elev,Range,Altitude);
```

```
        if (Altitude > TooLow) and (Altitude < TooHigh) then
            Ambig:= false;
    end;
    if Ambig and ( (360-Phas > 300) or (360-Phas < 20) ) then begin
       {Elevation between 167 and 176 degrees.  Range > 396.}
        if TestGraphicsOn then  PlotASymbol(Cross,Range,360 - Phas,2);
        Range:= Range + 396;
        Elev:= (Phas/(360) + 4.0) / AerialSpacing;
        if (Elev > 1) and (Elev < 1.1) then begin
            Elev:= 1;
            Ambig:= true;
        end else
            Elev:= InvCosine(Elev);
        Elev:= 180 - Elev;
        Get_Altitude(Elev,Range,Altitude);
        if (Altitude < TooHigh) and (Altitude > 10) then  Ambig:= false;
       {NOTE the altitude is probably in considerable doubt anyway.}
    end;
    if TestGraphicsOn then if Ambig then
        PlotASymbol(Plus,Range,Phas,4);

end;
end;                                                    {end of ResolvePhase}


function CheckOut
    (var ObsvBuff:Observation) : boolean;
    {Checks the observing code detected a meteor on all three channels.}
    var  HomeOK,NuttOK,SpitOK : boolean;
begin
CheckOut:= true;
with ObsvBuff.Nutt do if DetectSweep = IndexOne then
    CheckOut:= false;
with ObsvBuff.Spit do if DetectSweep = IndexOne then
    CheckOut:= false;
with ObsvBuff do if Home.Range = 0) or Nutt.Range = 0)
    or (Spit.Range = 0) then CheckOut:= false;
end;                                                {end of CheckOut}


procedure DeSpikeProfile
    (var V:Vector; Noise:integer);
    {De-spikes the amplitude profile.}
    var  i,Bound : integer;
         Last : integer;
         Expected : integer;
begin
V[1]:= Noise;  Last:= Noise;
Bound:= 30;
if Bound < 3*Noise then Bound:= 3*Noise;
for i:= 2 to 249 do begin
    if (V[i] < Last - Bound) or (V[i] > Last + Bound) then begin
        Expected:= (V[i-1] + V[i+1]) div 2;
        if (V[i] < Expected - Bound) or (V[i] > Expected + Bound) then
```

```
            V[i]:= byte(Expected);
    end;
    Last:= V[i];
end;
if (V[250] < Last - Bound) or (V[250] > Last + Bound) then
    V[250]:= Last;

end;                                                {end of DeSpikeProfile}

procedure WriteBuffer_Txt
    (var OutText:text; var Buffer:ReducedData);
    {Writes out the contents of the ReducedData buffer out to a text file.
     The routine adds a few '*'s between each reduced data record.  When the
     data text file is read this will check that none of the numerical
     values have overrun their fields.}
begin
with Buffer do begin
    writeln(OutText,Year:4,Month:4,Day:4,Hour:4,Min:4,Sec:4,Number:6,
                    PersTime:8);
    with Home do begin
        writeln(OutText,Noise:4,Max:5,Imax:4,Range:5,Rise:4,Decay:4,
                DiffHeight:8:2,Altitude:8:2,Velocity:8:2,NumHalfCycles:6);
    end;
    with Nutt do begin
        writeln(OutText,Noise:4,Max:5,Imax:4,Range:5,Rise:4,Decay:4,
                DiffHeight:8:2,Altitude:8:2,Velocity:8:2,NumHalfCycles:6);
    end;
    with Spit do begin
        writeln(OutText,Noise:4,Max:5,Imax:4,Range:5,Rise:4,Decay:4,
                DiffHeight:8:2,Altitude:8:2,Velocity:8:2,NumHalfCycles:6);
    end;
    writeln(OutText,Lag12:14:6,Lag13:14:6,Lag23:14:6);
    writeln(OutText,AvTin:9:4,AvTos:9:4,Phase:9:4,Elev:9:4,
                    AvFresVel:12:4,AtmosDecel:12:4);
    writeln(OutText,Vx:12:4,Vy:12:4,Vz:12:4,Azimuth:10:4,
                    Zenith:9:4,Speed:12:4);
    writeln(OutText,RightAsc:12:4,Declin:12:4,SpeedGeo:12:4,
                    DayNumber:6,LongSun:12:4,Long1950:12:4);
    writeln(OutText,Rm:6:4,Vmx:12:4,Vmy:12:4,Vmz:12:4,SpeedHelio:12:4,
                    LongAsc:9:4,LongAsc1950:9:4);
    writeln(OutText,Inclin:10:4,SemiLatRect:12:4,Perihel:12:4,Ecc:12:4,
                    SemiMajor:12:4);
    writeln(OutText,PeriDist:8:4,PeriLong:12:4,PeriLat:12:4);
    writeln(OutText,'*******');
end;
end;                                            {end of WriteBuffer_Txt}

procedure ReadBuffer_Txt
    (var InText:text; var Buffer:ReducedData);
    {Reads in the contents of a reduced data record from a text file
     written with the WriteBuffer_Txt procedure.}
    var  Dummy : char;
begin
with Buffer do begin
```

```
    readln(InText,Year,Month,Day,Hour,Min,Sec,Number,PersTime);
    with Home do begin
        readln(InText,Noise,Max,Imax,Range,Rise,Decay,
                    DiffHeight,Altitude,Velocity,NumHalfCycles);
    end;
    with Nutt do begin
        readln(InText,Noise,Max,Imax,Range,Rise,Decay,
                    DiffHeight,Altitude,Velocity,NumHalfCycles);
    end;
    with Spit do begin
        readln(InText,Noise,Max,Imax,Range,Rise,Decay,
                    DiffHeight,Altitude,Velocity,NumHalfCycles);
    end;
    readln(InText,Lag12,Lag13,Lag23);
    readln(InText,AvTin,AvTos,Phase,Elev,AvFresVel,AtmosDecel);
    readln(InText,Vx,Vy,Vz,Azimuth,Zenith,Speed);
    readln(InText,RightAsc,Declin,SpeedGeo,DayNumber,LongSun,Long1950);
    readln(InText,Rm,Vmx,Vmy,Vmz,SpeedHelio,LongAsc,LongAsc1950);
    readln(InText,Inclin,SemiLatRect,Perihel,Ecc,SemiMajor);
    readln(InText,PeriDist,PeriLong,PeriLat);
    readln(InText);
end;
end;                                          {end of ReadBuffer_Txt}
```

C.7 OrbLags

```
function LappLag
    (var Site1,Site2:SiteProfile) : integer;
    {Finds the time difference between the detection sweeps of the profiles.}
    var Lapp : integer;
begin
{If the observing code detected a meteor it will have put the relevant index
 into DetectedSweep and adjusted IndexOne accordingly.  If none is detected at
 the site then IndexOne is chosen by default and DetectedSweep set the same.}
Lapp:= integer(Site1.IndexOne) - integer(Site2.IndexOne);

{This next bit is a dangerous piece of programming since I intend to change
 the size of MaxSweep to something much larger.  Reprocessing old data will
 then be in danger.  I'll set a new constant ReducSweep in MetReduc.}
if (Site1.IndexOne > ReducSweep - 100) and (Site2.IndexOne < 100) then
    Lapp:= Lapp - ReducSweep;
if (Site1.IndexOne < 100) and (Site2.IndexOne > ReducSweep - 100) then
    Lapp:= Lapp + ReducSweep;

LappLag:= Lapp;
end;                                  {end of LappLag}


procedure FindMaxRise
    (Start,Finish:integer; var AmpDash:IntVector; var MaxDash:integer;
     var MaxRise:real; Noise,Max:byte; var RiseTime:byte; var Error:word);
    {Find point of max rising slope (MaxRise) and the rise time to the peak.}
    label  888;
```

```
    const  CutLevel = 0.70;
    var  i,j,k,m,n : integer;
         Icut : integer;
         RiseWidth : integer;
         Dummy : string;
         SlopeCut : real;
         Sum,X,Y,Z,D : real;
         Slope : real;
begin
MaxDash:= 0;  k:= 0;
if Finish > 235 then Finish:= 235;

for i:= Start to Finish do if AmpDash[i] > MaxDash then begin
    MaxDash:= AmpDash[i];
    k:= i;
end;

{Cannot do parabolic fit centred on (k,MaxDash).}
if (k < Start + 5) or (k > Finish - 5) then begin
    Error:= 17;  MaxRise:= 999.9;  goto 888;
end;

SlopeCut:= CutLevel*MaxDash;
i:= Start;  j:= Finish;
while AmpDash[i] < SlopeCut do i:= i + 1;
while AmpDash[j] < SlopeCut do j:= j - 1;
RiseWidth:= j - i + 1;
Icut:= (i + j);

{Check that there are values below 0.70*max before and after max.}
SlopeCut:= CutLevel*MaxDash;
if (AmpDash[Start] > SlopeCut) or (AmpDash[Finish] > SlopeCut) then begin
    Error:= 17;  MaxRise:= 999.9;  goto 888;
end;

{Now do a parabolic interpolation}
Sum:= 0.00;  Y:= MaxDash;
for i:= 1 to 5 do begin
    X:= AmpDash[k-i];
    Z:= AmpDash[k+i];
    D:= 0.5 - ((X-Y)/(X-(2*Y)+Z));
    Sum:= Sum + (D*i);
end;
MaxRise:= k - (Sum/5);

if TestGraphicsOn then begin
   {plot on the mid point of stuff above 0.70%}
    SetTextJustify(LeftText,CenterText);
    PlotPoint(Icut,20,Cross,4);
    PlotPoint(300,10,Plus,2);
    OutTextXY(307,10,'-');
    PlotPoint(320,10,Cross,2);
    str(abs(MaxRise - Icut/2):4:1,Dummy);
    OutTextXY(330,10,Dummy);
```

```
end;

{Ensure that the rising edge slope is not double peaked.}
if abs(MaxRise - Icut/2) >= 6 then begin
    Error:= 17;  MaxRise:= 888.8;  goto 888;
end;

{Calculate the rise time of the profile.}
Slope:= MaxDash / 300;
RiseTime:= round((Max - Noise) / Slope);

888:
end;                                                    {end of FindMaxRise}


procedure MaxRiseSlopeLags
    (var SmoothBuff:Observation; var Buffer:ReducedData;
     var Max12,Max13,Max23:real; var Error:word);
    {Finds timelags from the maximum rising slope points.}
    var   HomePt,NuttPt,SpitPt : real;
          Lapp12,Lapp13,Lapp23 : integer;
          MaxDash : integer;
          MaxRise : real;
          AmpDash : IntVector;
begin
with SmoothBuff do begin
    Lapp12:= LappLag(Home,Nutt);
    Lapp13:= LappLag(Home,Spit);
    Lapp23:= LappLag(Nutt,Spit);
end;

with Buffer.Home do begin
    DifferentiateProfile(SmoothBuff.Home,16,Imax + 5,AmpDash);
    FindMaxRise(16,Imax + 5,AmpDash,MaxDash,HomePt,Noise,Max,Rise,Error);
end;
with Buffer.Nutt do begin
    DifferentiateProfile(SmoothBuff.Nutt,16,Imax + 5,AmpDash);
    FindMaxRise(16,Imax + 5,AmpDash,MaxDash,NuttPt,Noise,Max,Rise,Error);
end;
with Buffer.Spit do begin
    DifferentiateProfile(SmoothBuff.Spit,16,Imax + 5,AmpDash);
    FindMaxRise(16,Imax + 5,AmpDash,MaxDash,SpitPt,Noise,Max,Rise,Error);
end;

if Error = 0 then begin
    Max12:= HomePt - NuttPt + Lapp12;
    Max13:= HomePt - SpitPt + Lapp13;
    Max23:= NuttPt - SpitPt + Lapp23;
end else begin
    Max12:= 999.9;  Max13:= 999.9;  Max23:= 999.9;
end;
end;                                          {end of MaxRiseSlopeLags}
```

C.8 OrbFres

```
procedure SketchMeanAmp
    (Imax,Max:byte; var SmoothRaw,MeanAmp:SiteProfile);
    {Sketches out a mean amplitude level.}
    var  i,j,Ilast : integer;
         Index : integer;          {Location of crossing point}
         Amp : integer;            {Running min or max amplitude}
         Alast : integer;          {Previous max or min amplitude}
         Step,LastStep,NumSteps : integer;
         Drop : real;              {Amp drop per index step}
         Mean : SiteProfile;
         SwapToSmooth : boolean;
begin
for i:= 1 to 250 do Mean.Profile[i]:= 0;
i:= Imax + 1;  Amp:= Max;
while (i < 250) do begin
   {Looking at the downward oscillation.}
    Ilast:= i - 1;  Alast:= Amp;
    while (SmoothRaw.Profile[i] <= Amp) and (i < 250) do begin
        Amp:= SmoothRaw.Profile[i]; i:= i + 1;
    end;
    Index:= Ilast + (i - Ilast - 1) div 2;
    Mean.Profile[Index]:= (Amp + Alast) div 2;

   {Looking at the upward oscillation.}
    Ilast:= i - 1;  Alast:= Amp;
    while (SmoothRaw.Profile[i] >= Amp) and (i < 250) do begin
        Amp:= SmoothRaw.Profile[i]; i:= i + 1;
    end;
    Index:= Ilast + (i - Ilast - 1) div 2;
    Mean.Profile[Index]:= (Amp + Alast) div 2;
end;

{Linear fit between the crossing points.}
i:= Imax;
while (Mean.Profile[i] = 0) and (i < 250) do i:= i + 1;
j:= i - 1;
while (j > Imax) do begin
    Mean.Profile[j]:= Mean.Profile[i];
    j:= j - 1;
end;
while (SmoothRaw.Profile[j] > Mean.Profile[i]) and (j > 45) do begin
    Mean.Profile[j]:= Mean.Profile[i];
    j:= j - 1;
end;
LastStep:= 200;  NumSteps:= 1;
SwapToSmooth:= false;
while (i < 250) and not SwapToSmooth do begin
    if TestGraphicsOn then
        PlotPoint(2*i,PlotShapeY(Mean.Profile[i]),Plus,4);
    Alast:= Mean.Profile[i];  Ilast:= i;
    i:= i + 1;
    while (Mean.Profile[i] = 0) and (i < 250) do i:= i + 1;
```

```
    Step:= 1;  Drop:= (Alast - Mean.Profile[i]) / (i - Ilast);
    for j:= Ilast + 1 to i - 1 do begin
        Mean.Profile[j]:= Alast - round(Step*Drop);
        Step:= Step + 1;
    end;
    if (Step > LastStep + 2) and (NumSteps > 3) then
        SwapToSmooth:= true;
    LastStep:= Step;  NumSteps:= NumSteps + 1;
end;
while (i < 250) do begin
    Mean.Profile[i]:= SmoothRaw.Profile[i];
    i:= i + 1;
end;
TriSmooth(Mean,MeanAmp,5);
end;                                                    {end of SketchMeanAmp}


procedure SizeHalfCycles
    (var Residue:IntVector; Istart:integer; var CycleSize:HalfCycleSizes);
    {Determines the number of pulses in each half cycle.}
    var   Ilast : integer;
          i,j : word;
          Cycle : integer;
begin
i:= Istart;  Cycle:= -1;
for j:= 0 to 29 do begin          {number of full cycles}
    Ilast:= i;  Cycle:= Cycle + 1;
    while (Residue[i] >= 0) and (i < 250 ) do  i:= i + 1;
    CycleSize[Cycle]:= i - Ilast;
    Ilast:= i;  Cycle:= Cycle + 1;
    while (Residue[i] < 0) and (i < 250 ) do i:= i + 1;
    CycleSize[Cycle]:= i - Ilast;
end;
end;                                                  {end of SizeHalfCycles}

procedure FindEndCycle
    (var Residue:IntVector; Istart:integer; var Iend:integer;
     var CycleSize:HalfCycleSizes; var EndCycle:shortint);
    {Finds the end of the oscillations based on amplitude and relative
     length of the particular half cycle.}
    var   i,j,Cycle : integer;
          LastTwo,ThisTwo : integer;        {Size of last full cycles}
          smax : word;
begin
Cycle:= 4;
LastTwo:= CycleSize[Cycle - 2] + CycleSize[Cycle - 3];
ThisTwo:= CycleSize[Cycle] + CycleSize[Cycle - 1];

while ((ThisTwo < LastTwo + 2) or (CycleSize[Cycle] < 2)) and
    (Cycle < 59) do begin
    Cycle:= Cycle + 1;
    LastTwo:= CycleSize[Cycle - 2] + CycleSize[Cycle - 3];
    ThisTwo:= CycleSize[Cycle] + CycleSize[Cycle - 1];
end;
```

```
EndCycle:= Cycle - 1;

i:= Istart;  Cycle:= 0;  smax:= 1000;
while (smax > 4) and (Cycle < 60) do begin
    smax:= 0;
    for j:= 1 to CycleSize[Cycle] do begin
        if abs(Residue[i]) > smax then smax:= abs(Residue[i]);
        i:= i + 1;
    end;
    Cycle:= Cycle + 1;
end;
Cycle:= Cycle - 2;
if EndCycle > Cycle then EndCycle:= Cycle;

i:= 0;
for Cycle:= 0 to EndCycle do i:= i + CycleSize[Cycle];
Iend:= Istart + i - 1;

if EndCycle < 6 then EndCycle:= -4;
if Iend - (Istart + CycleSize[0]) < 20 then EndCycle:= - 4;

end;                                                    {end of FindEndCycle}


procedure GetInvSinY
    (var CycleSize:HalfCycleSizes; Istart,EndCycle:integer;
     var Residue:IntVector; var InvSinY:RealVector);
    {Returns the inverse sine of the normalised oscillations.}
    var   i,j,Ilast : word;
          Cycle : integer;
          smax : integer;
          NormRes : RealVector;
          AddPhase : real;
          Sign : integer;
          FirstMax : boolean;
begin
i:= Istart + CycleSize[0];
Sign:= -1;  AddPhase:= Pi;
for Cycle:= 1 to EndCycle do begin

    FirstMax:= true;
    Ilast:= i; smax:= 0;
    for j:= 1 to CycleSize[Cycle] do begin
        if abs(Residue[i]) > smax then smax:= abs(Residue[i]);
        i:= i + 1;
    end;
    if (smax = 0) then  NormRes[Ilast]:= 1  else
        for j:= Ilast to i - 1 do NormRes[j]:= Residue[j] / abs(smax);

    for j:= Ilast to i - 1 do begin
        InvSinY[j]:= Pi*InvSine(NormRes[j]) / 180;
        InvSinY[j]:= AddPhase + Sign*InvSinY[j];

        if (abs(NormRes[j]) = 1.0) then
```

```
            if FirstMax then begin
                AddPhase:= AddPhase + Pi;
                Sign:= - Sign;
                FirstMax:= false;
            end else
                InvSinY[j]:= AddPhase - (Pi / 2);
    end;
end;
end;                                                    {end of GetInvSinY}


procedure FitVelocity
    (FirstPoint,Iend:integer; var InvSinY:RealVector; Range:real;
     var Tzero,Velocity:real);
    {Least squares fit to determine a Fresnel velocity.}
    var   x,y,y2 : real;
          m,c : real;
          Num : integer;
          Sx,Sy,Sxx,Sxy,Syy : real;
          t : integer;
begin
Num:= 0;  Sx:= 0;  Sy:= 0;  Sxx:= 0;  Sxy:= 0;  Syy:= 0;
for t:= FirstPoint to Iend do begin
        x:= t / Prf;
        y2:= InvSinY[t];
        y:= sqrt(y2);
        Num:= Num + 1;  Sx:= Sx + x;  Sy:= Sy + sqrt(y2);
        Sxx:= Sxx + sqr(x);  Sxy:= Sxy + x*y;  Syy:= Syy + y2;
    end;

m:= (Sx*Sy - Num*Sxy) / (Sx*Sx - Num*Sxx);
c:= (Sy*Sxx - Sx*Sxy) / (Num*Sxx - Sx*Sx);

Velocity:= sqr(m)*Range*(0.001*WaveLength) / (2*Pi);
Velocity:= sqrt(Velocity);
end;                                                    {end of FitVelocity}


procedure FresnelVelocity
    (var Smoothed,Site:SiteProfile; Range:real; var EndCycle:shortint;
     var VelSubDomain:RealVector; var Velocity:real);
    {Gets the Fresnel velocity and deceleration for one amplitude profile.}
    label 777;
    var   Istart,Iend : integer;
          MeanAmp : SiteProfile;
          CycleSize : HalfCycleSizes;
          Residue : IntVector;
          InvSinY : RealVector;
          Cycle : integer;
          i,j,k : integer;
          t : integer;
          FirstPoint : word;
          Imax,Max : byte;
          Tzero,Beta : real;
```

```
        FresDecel : real;
begin
{This zeroing is probably conservative.}
for i:= 1 to 250 do InvSinY[i]:= 0.00;
FindMax(Smoothed,Imax,Max,100);

{Search back for a missed earlier maximum}
i:= Imax;
while (i > 40) and (Smoothed.Profile[i] > Max - 24) do i:= i - 1;
while (i > 40) and (Smoothed.Profile[i] < Max - 10) do i:= i - 1;
if i > 40 then FindMax(Smoothed,Imax,Max,i);

SketchMeanAmp(Imax,Max,Smoothed,MeanAmp);
for i:= 1 to 250 do Residue[i]:= Site.Profile[i] - MeanAmp.Profile[i];

Istart:= Imax;
SizeHalfCycles(Residue,Istart,CycleSize);
FindEndCycle(Residue,Istart,Iend,CycleSize,EndCycle);
if EndCycle = -4 then goto 777;

GetInvSinY(CycleSize,Istart,EndCycle,Residue,InvSinY);
GetFresnelDecel(Istart,Iend,EndCycle,InvSinY,CycleSize,
                Range,VelSubDomain,Beta,FresDecel);

if (EndCycle = -5) then begin
    Velocity:= 0.0;
end else
    FitVelocity(Istart + CycleSize[0],Iend,InvSinY,Range,Tzero,Velocity);

777:
end;                                                {end of FresnelVelocity}


procedure GetAtmosDecel
    (var AllSub,AllTime:RealVector; NumAll:integer; var Beta,AtmosDecel:real);
    {Fits an atmospheric deceleration to composite velocity sub domain data.}
    var   i : integer;
          t,Vel : real;
          Count,Cycle : integer;

          SumX,SumY : real;
          SumSqrX,SumXY : real;
          Num : word;
          MeanX,MeanY : real;
          Sxx,Sxy,Syy : real;
begin
SumX:= 0 ;  SumY:= 0;  SumXY:= 0;  SumSqrX:= 0; Num:= 0;
for i:= 1 to NumAll do begin
    t:= AllTime[i] / Prf;
    SumX:= SumX + t;
    SumSqrX:= SumSqrX + sqr(t);
    SumY:= SumY + AllSub[i];
    SumXY:= SumXY + t*AllSub[i];
    Num:= Num + 1;
```

```
end;

if Num > 8 then begin
    MeanX:= SumX / Num;
    MeanY:= SumY / Num;
    Sxx:= SumSqrX - sqr(SumX) / Num;
    Sxy:= SumXY - SumX*SumY/Num;

    if Sxx = 0 then Sxx:= 1.0e-10;
    AtmosDecel:= - Sxy/Sxx;
    Beta:= MeanY + MeanX*AtmosDecel;
end else AtmosDecel:= 999;
end;                                                    {end of GetAtmosDecel}


procedure AverageVelocities
    (V1,V2,V3:real; W1,W2,W3:integer; var NumFres:integer; var AvVel:real);
    {Finds the average Fresnel velocity.}
    label   777;
    var  U : array[1..3] of real;
         Y : array[1..3] of real;
         D1,D2,D3 : real;
         i : integer;
begin
i:= 0;
if V1 > 0 then begin
    i:= i + 1;  U[i]:= V1;  Y[i]:= W1;
end;
if V2 > 0 then begin
    i:= i + 1;  U[i]:= V2;  Y[i]:= W2;
end;
if V3 > 0 then begin
    i:= i + 1;  U[i]:= V3;  Y[i]:= W3;
end;

NumFres:= i;  AvVel:= 0;
for i:= 1 to NumFres do AvVel:= AvVel + U[i];
if NumFres = 0 then goto 777;

AvVel:= AvVel / NumFres;
if NumFres = 1 then goto 777;

D1:= abs(AvVel - U[1]);  D2:= abs(AvVel - U[2]);  D3:= abs(AvVel - U[3]);
if NumFres = 3 then  if (D1 > 5) or (D2 > 5) or (D3 > 5) then begin
    if (D3 < D1) or (D3 < D2) then if D1 > D2 then begin
        U[1]:= U[3];  Y[1]:= Y[3];
    end else begin
        U[2]:= U[3];  Y[2]:= Y[3];
    end;
    AvVel:= (U[1] + U[2]) / 2;
    NumFres:= 2;
end;

if NumFres = 2 then
```

```
if (abs(AvVel - U[1]) > 5) or (abs(AvVel - U[2]) > 5) then begin
    if Y[1] > Y[2] then AvVel:= U[1]
        else AvVel:= U[2];
    NumFres:= 1;
end;
777:
end;                                                {end of AverageVelocities}


procedure GetFresnelVelocities
    (var ObsvBuff,SmoothBuff:Observation; ShowPlot:boolean;
     var Buffer:ReducedData);
    {Provides the shell for the Fresnel Velocity calculations.}
    label  777;
    var  HomeSub,NuttSub,SpitSub : RealVector;
         NumAll : integer;
         AllSub,AllTime : RealVector;
         FirstPoint,LastPoint : integer;
         i,j : integer;
         Lapp12,Lapp13,Lapp23 : integer;
         HomeLag,NuttLag,SpitLag : integer;
         Vel,VelLow : integer;
         Beta : real;
         DecelVal : string;
         NumFres : integer;
begin
if ShowPlot then begin
    SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);
    ClearViewPort;
    SetTextJustify(LeftText,CenterText);
    OutTextXY(20,22,'Processing Observation');
    with Buffer do OutTextXY(20,34,Datestr(Year,Month,Day)
             +'   '+ Timestr(Hour,Min,Sec) +'   '+ intstr(Number));
end;

with ObsvBuff do begin
    Lapp12:= LappLag(Home,Nutt);
    Lapp13:= LappLag(Home,Spit);
    Lapp23:= LappLag(Nutt,Spit);
end;
if (Lapp12 < 0) and (Lapp13 < 0) then begin
    HomeLag:= 0;  NuttLag:= - Lapp12;  SpitLag:= - Lapp13;
end;
if (Lapp12 >= 0) and (Lapp23 < 0) then begin
    HomeLag:= Lapp12;  NuttLag:= 0;  SpitLag:= - Lapp23;
end;
if (Lapp13 >= 0) and (Lapp23 > 0) then begin
    HomeLag:= Lapp13;  NuttLag:= Lapp23;  SpitLag:= 0;
end;

NumAll:= 0;
with Buffer.Home do  FresnelVelocity(SmoothBuff.Home,ObsvBuff.Home,
        Range,NumHalfCycles,HomeSub,Velocity);
if Buffer.Home.Velocity > 0 then  for i:= 1 to 250 do
```

```
    if (HomeSub[i] > 0) then begin
        NumAll:= NumAll + 1;
        AllSub[NumAll]:= HomeSub[i];
        AllTime[NumAll]:= i + HomeLag;
    end;

with Buffer.Nutt do  FresnelVelocity(SmoothBuff.Nutt,ObsvBuff.Nutt,
        Range,NumHalfCycles,NuttSub,Velocity);
if Buffer.Nutt.Velocity > 0 then  for i:= 1 to 250 do
    if (NuttSub[i] > 0) then begin
        NumAll:= NumAll + 1;
        AllSub[NumAll]:= NuttSub[i];
        AllTime[NumAll]:= i + NuttLag;
    end;

with Buffer.Spit do  FresnelVelocity(SmoothBuff.Spit,ObsvBuff.Spit,
        Range,NumHalfCycles,SpitSub,Velocity);
if Buffer.Spit.Velocity > 0 then  for i:= 1 to 250 do
    if (SpitSub[i] > 0) then begin
        NumAll:= NumAll + 1;
        AllSub[NumAll]:= SpitSub[i];
        AllTime[NumAll]:= i + SpitLag;
    end;

with Buffer do AverageVelocities(Home.Velocity,Nutt.Velocity,Spit.Velocity,
  Home.NumHalfCycles,Nutt.NumHalfCycles,Spit.NumHalfCycles,NumFres,AvFresVel);
if NumFres = 0 then goto 777;

if ShowPlot then begin
    VelLow:= round(Buffer.AvFresVel - 10);
    DrawDecelScales(Buffer,VelLow);
    if Buffer.Home.NumHalfCycles > 0 then
        PlotVelSubDomain(VelLow,HomeLag,HomeSub,Cross);
    if Buffer.Nutt.NumHalfCycles > 0 then
        PlotVelSubDomain(VelLow,NuttLag,NuttSub,FillBox);
    if Buffer.Spit.NumHalfCycles > 0 then
        PlotVelSubDomain(VelLow,SpitLag,SpitSub,Spot);

{Test bit to check I got the right points !!}
{   for i:= 1 to NumAll do begin
        Vel:= 200 - round(10*(AllSub[i] - VelLow));
        PlotPoint(3*(round(AllTime[i]) - 50),Vel,Plus,4);
    end;
}
end;

if NumFres < 2 then begin
    Buffer.AtmosDecel:= 999.9;
    if ShowPlot then begin
        SetTextJustify(LeftText,CenterText);
        OutTextXY(-40,-20,'Only one Fresnel fit')
    end;
    goto 777;
end;
```

```
GetAtmosDecel(AllSub,AllTime,NumAll,Beta,Buffer.AtmosDecel);

if ShowPlot then begin
    FirstPoint:= 250;  LastPoint:= 1;
    for i:= 1 to NumAll do begin
        if AllTime[i] < FirstPoint then FirstPoint:= round(AllTime[i]);
        if AllTime[i] > LastPoint then LastPoint:= round(AllTime[i]);
    end;

    SetTextJustify(LeftText,CenterText);
    if Buffer.AtmosDecel = 999 then OutTextXY(-40,-20,'No Decel fit')
    else begin
        str(Buffer.AtmosDecel:6:1,DecelVal);
        OutTextXY(-40,-20,'Decel '+ DecelVal +' km.s-2');
    end;

    i:= FirstPoint - 5;
    Vel:= 200 - round(10*(Beta - Buffer.AtmosDecel*i/Prf - VelLow));
    MoveTo(3*(i - 50),Vel);
    i:= LastPoint + 5;
    Vel:= 200 - round(10*(Beta - Buffer.AtmosDecel*i/Prf - VelLow));
    LineTo(3*(i - 50),Vel);
end;

777:
end;                                          {end of GetFresnelVelocities}
```

C.9 OrbElems

```
procedure SetMatricies
    (var DaysInYear:DaysInYearArray; var A1,T,A2,A3,A4:TransMatrix);
    {Initialise the translation matrix elements.}
    var  i,j : integer;
begin
DaysInYear[89]:= 365;  DaysInYear[93]:= 365;  DaysInYear[97]:= 365;
DaysInYear[90]:= 366;  DaysInYear[94]:= 365;  DaysInYear[98]:= 365;
DaysInYear[91]:= 365;  DaysInYear[95]:= 365;  DaysInYear[99]:= 365;
DaysInYear[92]:= 366;  DaysInYear[96]:= 366;

{Matrix for rotation to the north celestial pole.  Latitude = -43.825 N}
    A1[1,1]:=  0.72146;    A1[1,2]:=  0;    A1[1,3]:= -0.69246;
    A1[2,1]:=      0  ;    A1[2,2]:=  1;    A1[2,3]:=      0  ;
    A1[3,1]:=  0.69246;    A1[3,2]:=  0;    A1[3,3]:=  0.72146;

{Transfer the axes so z axis points to the north celestial pole.}
    T [1,1]:=  0;    T [1,2]:=  1;    T [1,3]:=  0;
    T [2,1]:=  0;    T [2,2]:=  0;    T [2,3]:=  1;
    T [3,1]:=  1;    T [3,2]:=  0;    T [3,3]:=  0;

{Point the x axis toward the vernal equinox.}
for i:= 1 to 3 do for j:= 1 to 3 do A2[i,j]:= 0.00;

{Rotate to ecliptic coords.  Inclination = 23.45 deg, 23.441 might be good}
```

```
    A3[1,1]:=   1 ;    A3[1,2]:=  0.00000;    A3[1,3]:=  0.00000;
    A3[2,1]:=   0 ;    A3[2,2]:=  0.91741;    A3[2,3]:=  0.39795;
    A3[3,1]:=   0 ;    A3[3,2]:= -0.39795;    A3[3,3]:=  0.91741;

{Rotate to heliocentric coords.  Need to calculate the solar longitude.}
for i:= 1 to 3 do for j:= 1 to 3 do A4[i,j]:= 0.00;

end;                                                    {end of SetMatricies}

procedure Mul_AxV
    (A:TransMatrix; V:ThreeVector; var Y:ThreeVector);
    {Routine to multiply the 3x3 matrix A by the 3 vector V.
               --- j ------>
           |     11   12   13
           |     21   22   23      for i x j matrix A, A[i,j]
        i  |     31   32   33
           v
     The routine was tested by TMatMul.Pas with a number of matrices.}

    var  i,j : integer;
begin

for i:= 1 to 3 do begin
    Y[i]:= 0;
    for j:= 1 to 3 do
        Y[i]:= Y[i] + A[i,j]*V[j];
end;
end;                                                    {end of Multiply_AxV}


function GtZero
    (Value : real) : real;
    {Returns a value which is 1.0E-9 if the input is less than that. The
     sign of the value is preserved.}
    var  Sign : real;
begin
if Value < 0.00 then Sign:= -1.0 else Sign:= 1.0;
if abs(Value) < 1.0E-9 then
    GtZero:= Sign*1.0E-9
else
    GtZero:= Value;
end;                                                        {end of GtZero}

function RMOD
    (Value,Denom : real) : real;
    {Returns the remainder after the Value/Denom.  Similar to mod but for
     real operands.}
begin
while Value > Denom do Value:= Value - Denom;
while Value < 0.00 do Value:= Value + Denom;
RMOD:= Value;
end;                                                        {end of RMOD}
```

```
function DayOfYear
    (Year,Month,Day:integer) : integer;
    {Calculates the day of the year.  Should work up to the year 2000.}
    var   DaysInMonth : array[1..12] of integer;
          Days,i : integer;
begin
DaysInMonth[1]:= 31;  DaysInMonth[5]:= 31;  DaysInMonth[9] := 30;
DaysInMonth[2]:= 28;  DaysInMonth[6]:= 30;  DaysInMonth[10]:= 31;
DaysInMonth[3]:= 31;  DaysInMonth[7]:= 31;  DaysInMonth[11]:= 30;
DaysInMonth[4]:= 30;  DaysInMonth[8]:= 31;  DaysInMonth[12]:= 31;
Days:= 0;
for i:= 1 to Month - 1 do
    Days:= Days + DaysInMonth[i];
if (Year mod 4 = 0) and (Year <> 100) and (Month > 2) then
    Days:= Days + 1;
Days:= Days + Day;

DayOfYear:= Days;
end;                                                    {end of DayOfYear}


function CheckAzimuthOK
    (Lag12,Lag13,Lag23,Azimuth:real) : boolean;
    {NOTE  I am worried about this routine in case it does not deal very
           well with the boundaries between the sectors.  What about
           small changes in values shifting an azimuth across a boundary.}

    {Thinking about it there is no way this routine should pick up any
     errors unless the time lags are all very small (?!).  The fact is
     it has picked up a couple of programing errors so why not leave it
     in for the moment!}

    {Check that the Azimuth angle of the meteor is consistent with timelags.}
    var   LooksOK : boolean;
begin
LooksOK:= false;
if (Lag12 >= 0) and (Lag13 >= 0) and (Lag23 < 0) then
    if (Azimuth >= 90 - Beta) and (Azimuth <= 90) then  LooksOk:= true;
if (Lag12 >= 0) and (Lag13 >= 0) and (Lag23 >= 0) then
    if (Azimuth >= 90) and (Azimuth <= 90 + Alpha) then  LooksOk:= true;
if (Lag12 < 0) and (Lag13 >= 0) and (Lag23 >= 0) then
    if (Azimuth >= 90 + Alpha) and (Azimuth <= 90+Beta) then LooksOk:= true;
if (Lag12 < 0) and (Lag13 < 0) and (Lag23 >= 0) then
    if (Azimuth >= 90 + Beta) and (Azimuth <= 270) then  LooksOk:= true;
if (Lag12 < 0) and (Lag13 < 0) and (Lag23 < 0) then
    if (Azimuth >= 270) and (Azimuth <= 270 + Alpha) then  LooksOk:= true;
if (Lag12 >= 0) and (Lag13 < 0) and (Lag23 < 0) then begin
    if (Azimuth >= 270 + Alpha) then  LooksOk:= true;
    if (Azimuth < Beta - 90) then  LooksOk:= true;
end;

CheckAzimuthOK:= LooksOK;
end;                                                {end of CheckAzimuth}
```

```
procedure GetOrbit
    (OutName:string; var Buffer:ReducedData; var Error:word);
    {This is the main procedure of this code unit.}

label  999;

var {Days in each year until the year 2000 AD}
       DaysInYear : DaysInYearArray;
    {Matrix for rotation to the north celestial pole.  Latitude = -43.825 N}
       A1 : TransMatrix;
    {Transfer the axes so z axis points to the north celestial pole.}
        T : TransMatrix;
    {Point the x axis toward the vernal equinox.}
       A2 : TransMatrix;
    {Rotate to ecliptic coords. Inclination = 23.45deg, 23.441 might be good}
       A3 : TransMatrix;
    {Rotate to heliocentric coords.  Need to calculate the solar longitude.}
       A4 : TransMatrix;

       X2,X3 : real;                {Velocity projections.  (s.m-1) }
       Elevation : real;            {radian Buffer.Elev}
       tanA,tanz : real;            {Calculation variables}

       Azimuth : real;              {Azimuth of the observed meteor}
       Zenith : real;               {Zenith angle of the meteor trail}
       Speed1 : real;               {Scalar speed of the meteor}
       V1 : ThreeVector;            {Observed velocity vector of meteor}
       Speed2 : real;               {Speed before entering the atmosphere}
       V2 : ThreeVector;            {Velocity before entering atmosphere}
       V3 : ThreeVector;            {Velocity from a non-rotating Earth}
       Speed3 : real;               {Final speed in Earth's grav. field}
       Speed3Sqr : real;
       Speed4 : real;               {Unperturbed speed of meteoroid}
       DeltaZ : real;               {Zenith attraction angle}
       V4 : ThreeVector;            {Velocity unperturbed by gravity}
       Y1,Y2 : ThreeVector;         {Dummies to rotate to equatorial}
       V5 : ThreeVector;            {Velocity in Equatorial Coordinates}
       RA : real;                   {Right ascension and declination of the}
       Dec : real;                  {    corrected radiant point}
       MJD : real;                  {Modified Julian date}
       NZST : real;                 {New Zealand standard time  (hours) }
       UT : real;                   {Universal time of observation  (hours) }
       DaysSince89 : integer;       {Days since 0 Jan 1989}
       JulianCent : real;
       GMST : real;                 {Greenwich mean sidereal time  (hours) }
       BFMST : real;                {Local sidereal time, Birdlings Flat}
       RotAngle : real;             {radian rotation angle}
       Num : real;
       MS : real;                   {Sun's mean anomaly}
       RS : real;                   {Earth to Sun distance  (au) }
       LongSun : real;              {Longitude of the Sun }
       Precession : real;           {Precession from mean equinox of 1950.0 }
       V6 : ThreeVector;            {Velocity in Heliocentric Coordinates}
       WE : real;                   {Earth's longitude of perihelion}
```

```
        ME : real;                      {Earth's mean anomaly}
        TE : real;                      {Earth's true anomaly}
        RE : real;                      {Earth's orbital radius}
        VEarth : ThreeVector;           {Earth's heliocentric velocity}
        SpeedEarth : real;              {Earth's orbital speed}
        DevApex : real;
           {Deviation of Earth's apex from perpendicular to solar longitude}
        Vm : ThreeVector;               {Unperturbed heliocentric velocity}
        Rm : real;                      {Heliocentric radius of velocity Vm }

        Vr : real;                      {Radial component of the velocity}
        Vnu2 : real;                    {Square of the angular component}
        tanNu : real;

        LongAsc : real;                 {Longitude of the ascending node}
        Inclin : real;                  {Inclination of the orbit}
        TrueAnomaly : real;             {True anomaly at observation}
        SemiLatRect : real;             {Semi-latus rectum of the meteor orbit}
        Perihel : real;                 {Argument of perihelion}
        Ecc : real;                     {Eccentricity of the orbit}
        SemiMajor : real;               {Semi-major axis}

        X1 : real;                      {dummy variable for calculation}
        Speed : real;                   {speed used for calculation}
        OutFile : text;                 {diagnostics output file}
        i : integer;

begin
if DiagnosticsOn then begin
    assign(OutFile,OutName);
    append(OutFile);
end;
Error:= 0;
SetMatricies(DaysInYear,A1,T,A2,A3,A4);

{First get the Azimuth, Zenith and Velocity of the meteor given the range
 elevation and timelags of the observation.}

X2:= Buffer.Lag12 / (D12 * Prf);
X3:= Buffer.Lag13 / (D13 * Prf);
tanA:= (X3*cosA - X2*cosB) / GtZero(-X3*sinA + X2*sinB);
Azimuth:= arctan(tanA);

{if the radar echo came from the north then the meteor must have been
 moving from the south to the north making   90 > Azimuth > 270. }
if Buffer.Elev > 90 then Azimuth:= Azimuth + Pi;
Azimuth:= RMOD(Azimuth,2*Pi);
Buffer.Azimuth:= RadToDeg*Azimuth;
with Buffer do
    if not CheckAzimuthOK(Lag12,Lag13,Lag23,Azimuth) then begin
       {Azimuth not consistant with time lags.}
        Error:= 22;
        goto 999;
    end;
```

```
Elevation:= Buffer.Elev * DegToRad;
tanz:= sin(Elevation) / GtZero(cos(Elevation)*cos(Azimuth));
Zenith:= arctan(tanz);
{The zenith angle will always be in the range  0 < zenith < 90 .}

{neither X2 or X3 should be near zero in the option chosen.}
if abs(X2) > abs(X3) then
    Speed1:= 0.5*sin(zenith)*cos(Alpha*DegToRad - Azimuth) / X2
else
    Speed1:= 0.5*sin(zenith)*cos(Azimuth - Beta*DegToRad) / X3;
Speed1:= abs(Speed1);
if Speed1 > 1000 then Speed1:= 1000.0;

V1[1]:= - Speed1 * sin(zenith) * cos(Azimuth);
V1[2]:= - Speed1 * sin(zenith) * sin(Azimuth);
V1[3]:= - Speed1 * cos(zenith);
with Buffer do begin
    Vx:= V1[1];  Vy:= V1[2];  Vz:= V1[3];
end;

if DiagnosticsOn then DisplayVector(OutFile,'Observed Velocity',V1);

Buffer.Zenith:= RadToDeg*Zenith;
Buffer.Speed:= Speed1;

{Atmospheric Deceleration.  Modified from 0.80, 7 November 1990.}
Speed2:= 1.6*ln(Speed1);
Speed2:= sqrt( sqr(Speed1) + 0.27*exp(Speed2) );
V2[1]:= - Speed2 * sin(zenith) * cos(Azimuth);
V2[2]:= - Speed2 * sin(zenith) * sin(Azimuth);
V2[3]:= - Speed2 * cos(zenith);
if DiagnosticsOn then DisplayVector(OutFile,'Velocity above Atmosphere',V2);

{Rotation of the Earth.}
V3:= V2;
V3[2]:= V2[2] + VRotation;
if DiagnosticsOn then DisplayVector(OutFile,'w.r.t a non-rotating Earth',V3);

{Acceleration in the Earth's gravitational field.
     The Azimuth and Zenith here are corrected for the Earth's rotation.}
Speed3Sqr:= sqr(V3[1]) + sqr(V3[2]) + sqr(V3[3]);
Speed3:= sqrt(speed3Sqr);
tanA:= V3[2] / GtZero(V3[1]);
Azimuth:= arctan(tanA);
if Buffer.Elev > 90 then Azimuth:= Azimuth + Pi;
tanz:= sqrt(sqr(V3[1]) + sqr(V3[2])) / GtZero(abs(V3[3]));
Zenith:= arctan(tanz);

{    Increase in Speed.}
if Speed3Sqr < 124 then begin
   {Observed velocity less than escape velocity.}
    Error:= 21;  goto 999;
end;
```

```
Speed4:= sqrt(Speed3Sqr - 123.3);
Buffer.SpeedGeo:= Speed4;

{    Zenith Attraction.}
tanz:= (Speed3 - Speed4) / (Speed3 + Speed4);
tanz:= tanz * sin(Zenith / 2) / cos(Zenith / 2);
DeltaZ:= 2 * arctan(tanz);
Zenith:= Zenith + DeltaZ;

V4[1]:= -Speed4 * sin(Zenith) * cos(Azimuth);
V4[2]:= -Speed4 * sin(Zenith) * sin(Azimuth);
V4[3]:= -Speed4 * cos(Zenith);

if DiagnosticsOn then begin
    DisplayVector(OutFile,'Unperturbed velocity',V4);
    writeln(OutFile,'corrected Azimuth':45,Azimuth*RadToDeg:7:2,'(deg)':7);
    writeln(OutFile,'corrected Zenith':45,Zenith*RadToDeg:7:2,'(deg)':7);
    writeln(OutFile,'unperturbed speed':45,Speed4:7:2,'(kms)':7);
    writeln(OutFile);
end;

{Coordinate Transformations.}
{    to equatorial coords}
Mul_AxV(A1,V4,Y1);
if DiagnosticsOn then  WriteOutMatrix(OutFile,
    A1,V4,Y1,'Rotate coordinate set to the north celestial pole');

Mul_AxV(T,Y1,Y2);
if DiagnosticsOn then  WriteOutMatrix(OutFile,
    T,Y1,Y2,'Point z axis to the north celestial pole');

{The time recorded at observation is New Zealand standard time.  This is
 converted to UT and the date to the modified Julian date.}
with Buffer do begin
    DayNumber:= DayOfYear(Year,Month,Day);
    NZST:= Hour + Min/60 + Sec/3600;
    UT:= NZST - 12;
    if UT < 0.00 then begin
        UT:= UT + 24;
        DayNumber:= DayNumber - 1;      {Assume never run on New Years Day !!}
    end;
    DaysSince89:= 0;
    for i:= (Year - 1) downto (89) do
        DaysSince89:= DaysSince89 + DaysInYear[i];
    DaysSince89:= DaysSince89 + DayNumber;
    MJD:= 47526.0 + DaysSince89 + UT/24;
end;
{This relation holds good for 1989, Astronomical Almanac 1989.}
JulianCent:= (MJD - 51544.5) / 36525;
GMST:= 6.69737456 + 2400.051336*JulianCent + 0.0000258622*sqr(JulianCent)
                                           + 1.002737909*UT;
GMST:= 6.6424454 + 0.0657098243*DaysSince89 + 1.00273791*UT;
GMST:= RMOD(GMST,24.00);
BFMST:= RMOD(GMST + BF_Longitude,24.00);
```

```
RotAngle:= Pi*(BFMST - 6)/12;
A2[1,1]:= cos(RotAngle);  A2[1,2]:= -sin(RotAngle);
A2[2,1]:= -A2[1,2];       A2[2,2]:= A2[1,1];
A2[3,3]:= 1;
Mul_AxV(A2,Y2,V5);
if DiagnosticsOn then  WriteOutMatrix(OutFile,
    A2,Y2,V5,'Rotate x axis to vernal equinox.');

{calculate the right ascension and declination as a check}
Speed:= sqrt(sqr(V5[1]) + sqr(V5[2]) + sqr(V5[3]));
RA:= arctan(V5[2] / GtZero(V5[1]));
if RA < 0.00 then RA:= RA + Pi;
if V5[2] > 0.00 then RA:= RA + Pi;
Dec:= arctan( - V5[3] / GtZero(sqrt(sqr(V5[1]) + sqr(V5[2]))) );
{The negative sign has to be in there since the x and y components are squared
 which removes their quadrant information.  The velocity points away from the
 radiant position.}
Buffer.RightAsc:= RadToDeg*RA;
Buffer.Declin:= RadToDeg*Dec;

if DiagnosticsOn then begin
    writeln(OutFile,'For the corrected radiant position');
    writeln(OutFile,'Speed':45,Speed:7:2,'(km/s)':7);
    writeln(OutFile,'Right ascension':45,RA*RadToDeg:7:2,'(deg)':7);
    writeln(OutFile,'Declination':45,Dec*RadToDeg:7:2,'(deg)':7);
    writeln(OutFile);
end;


{    to heliocentric coords}
Mul_AxV(A3,V5,Y1);
if DiagnosticsOn then  WriteOutMatrix(OutFile,
    A3,V5,Y1,'Rotate to ecliptic coords');


{Astronomical Almanac page C24              ********************* til 2000 ?}
Num:= -4018.5 + DaysSince89 + UT/24;
MS:= 357.528 + 0.9856003*Num;
MS:= DegToRad*RMOD(MS,360.0);
LongSun:= 280.460 + 0.9856474*Num + 1.915*sin(MS) + 0.020*sin(2*MS);
LongSun:= RMOD(LongSun,360.0);
RS:= 1.00014 - 0.01671*cos(MS) - 0.00014*cos(2*MS);

Rotangle:= LongSun - 90;  RotAngle:= DegToRad * RotAngle;
A4[1,1]:= cos(RotAngle);  A4[1,2]:= sin(RotAngle);
A4[2,1]:= -A4[1,2];       A4[2,2]:= A4[1,1];
A4[3,3]:= 1;
Mul_AxV(A4,Y1,V6);
if DiagnosticsOn then  WriteOutMatrix(OutFile,
    A4,Y1,V6,'Rotate to heliocentric coords');
Buffer.LongSun:= LongSun;

{Orbital Motion of the Earth}
```

```
{Astronomical Almanac, calculated from page C1 (+- 180 deg) ***** for 1989 }
WE:= 102.749176 + 0.0000470684*(DaysSince89 + UT/24);
TE:= RMOD(LongSun + 180 - WE , 360.0);
TE:= DegToRad * TE;
RE:= RS;
X1:= G * Msun;
X1:= X1 * 1.00e-9;                          {convert dimensions to km3/s2 }
X1:= X1*( 2.00/RE - 1.00 ) / OneAU;         {units  km2/s2 }
SpeedEarth:= sqrt(X1);

DevApex:= - EE * sin(TE) / (1 + EE*cos(TE) );
VEarth[1]:= SpeedEarth * cos(DevApex);
VEarth[2]:= SpeedEarth * sin(DevApex);
VEarth[3]:= 0;
if DiagnosticsOn then  DisplayVector(OutFile,
    'Orbital velocity of the Earth',VEarth);

Vm[1]:= V6[1] + VEarth[1];
Vm[2]:= V6[2] + VEarth[2];
Vm[3]:= V6[3] + VEarth[3];
if DiagnosticsOn then  DisplayVector(OutFile,
    'Heliocentric velocity of meteor',Vm);
Buffer.Vmx:= Vm[1];
Buffer.Vmy:= Vm[2];
Buffer.Vmz:= Vm[3];
Buffer.SpeedHelio:= sqrt( sqr(Vm[1]) + sqr(Vm[2]) + sqr(Vm[3]) );

{Heliocentric position of Meteor}
Rm:= RE;
Buffer.Rm:= Rm;

{Orbital Elements}
{The Longitude of the ascending node should be the long. of the earth from the
 sun.}
Inclin:= arctan(Vm[3]/GtZero(Vm[1]));
if Vm[3] > 0.00 then begin                  {at Ascending node}
    LongAsc:= RMOD(LongSun - 180,360.0);
    if Inclin < 0.00 then Inclin:= Inclin + Pi;   {retrograde}
end else begin                              {at Descending node}
    LongAsc:= LongSun;
    if Inclin < 0.00 then Inclin:= abs(Inclin) {prograde}
        else Inclin:= Pi - Inclin;              {retrograde}
end;
Buffer.LongAsc:= LongAsc;
Buffer.Inclin:= RadToDeg*Inclin;

{The solar longitude is for the mean equinox of date.  It is more usual for
 meteor work to use the mean equinox of 1950.  To calculate the precession
 from 1950.}
JulianCent:= (MJD - 33281.923) / 36534.22;
Precession:= 1.396319*JulianCent + 0.0003083*sqr(JulianCent);
Buffer.Long1950:= LongSun - Precession;
Buffer.LongAsc1950:= LongAsc - Precession;
```

```
Vr:= -Vm[2];
Vnu2:= sqr(Vm[1]) + sqr(Vm[3]);

SemiLatRect:= sqr(Rm)*Vnu2 / (G*Msun);        {AU^2.km^2.s^-1 / (stuff) }
SemiLatRect:= SemiLatRect*OneAU*1.00e9;       {convert units to AU}
tanNu:= Vr*1.00e3 * sqrt(SemiLatRect*OneAu*1.00e3/(G*Msun))
                  / GtZero(SemiLatRect/Rm - 1);
TrueAnomaly:= arctan(tanNu);
if TrueAnomaly < 0.00 then TrueAnomaly:= TrueAnomaly + Pi;
if Vr < 0.00 then
    TrueAnomaly:= TrueAnomaly + Pi;
Buffer.SemiLatRect:= SemiLatRect;

if Vm[3] >= 0.00 then begin
    if TrueAnomaly < Pi then Perihel:= - TrueAnomaly
        else Perihel:= 2*Pi - TrueAnomaly;
end else
    Perihel:= Pi - TrueAnomaly;
Buffer.Perihel:= RadToDeg*Perihel;

Ecc:= (SemiLatRect/Rm - 1) / GtZero(cos(TrueAnomaly));
Ecc:= abs(Ecc);
SemiMajor:= SemiLatRect / GtZero(1 - sqr(Ecc));
Buffer.Ecc:= Ecc;
Buffer.SemiMajor:= SemiMajor;
Buffer.PeriDist:= SemiMajor*(1 - Ecc);
Buffer.PeriLat:= ArcSine( Sin(Inclin)*Sin(Perihel) );
Buffer.PeriLong:= DegToRad*Buffer.LongAsc1950
                    + arctan( cos(Inclin)*sin(Perihel)/cos(Perihel) );
if cos(Perihel) < 0 then Buffer.PeriLong:= Buffer.PeriLong + Pi;


if DiagnosticsOn then  begin
    writeln(OutFile,'Speed w.r.t. Earth':45,
            sqrt(sqr(V6[1]) + sqr(V6[2]) + sqr(V6[3])):8:2,'(km/s)':7);
    writeln(OutFile,'Heliocentric speed':45,Buffer.SpeedHelio:8:2,'(km/s)':7);
    writeln(OutFile,'Meteor true anomaly':45,
                                        TrueAnomaly*RadToDeg:8:2,'(deg)':7);
    writeln(OutFile);
    writeln(OutFile,'Greenwich Day Number':45,Buffer.DayNumber:7);
    writeln(OutFile,'Days since 0 Jan 1989':45,DaysSince89:7);
    writeln(Outfile,'Universal Time':45,UT:8:2,'(hrs)':7);
    writeln(OutFile,'GMST':45,GMST:8:3,'(hrs)':6);
    writeln(OutFile,'BFMST':45,BFMST:8:3,'(hrs)':6);
    writeln(OutFile);
end;

999:

if DiagnosticsOn then  close(OutFile);
end;                                                    {end of GetOrbit}

{************************************************************************}
```

```
begin                          {initialisation section}

DiagnosticsOn:= false;
DegToRad:= Pi / 180;
RadToDeg:= 180 / Pi;

VRotation:= -0.4631*cos(BF_Latitude*DegToRad);

cosA:= cos(Alpha*DegToRad);
sinA:= sin(Alpha*DegToRad);
cosB:= cos(Beta*DegToRad);
sinB:= sin(Beta*DegToRad);

end.
```

Appendix D

Stamp Album

All of science is either physics or stamp collecting.

Ernest Rutherford

This appendix is included to show the cumulative distributions of the station data and orbital elements for all the meteors observed up to the current date. This includes a total of about 132 000 meteor orbits collected in the fifteen months between 1990 Febuary and 1991 May. They include all the display options available in the `DistLot.Pas` program. Quite by chance the appendix has been labelled Appendix D for Data Distributions.

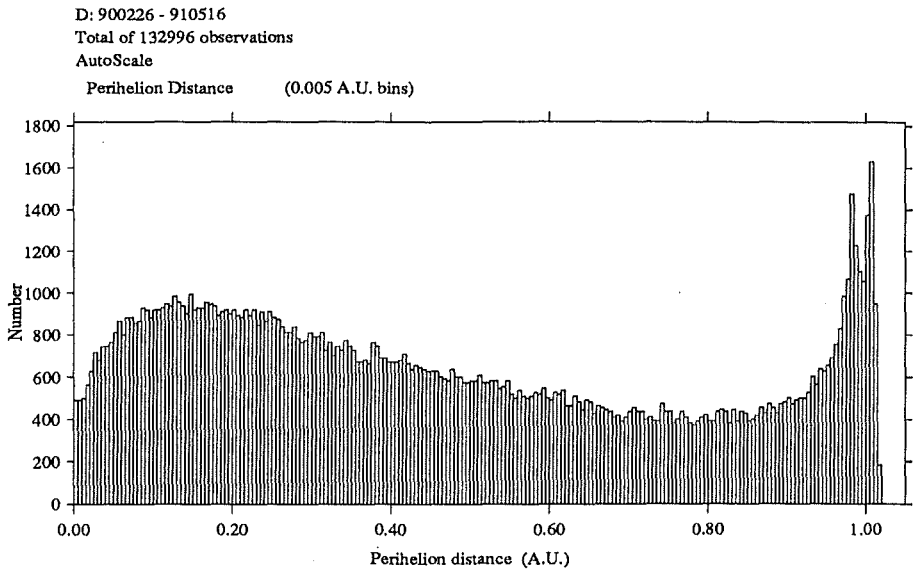


Figure D.1: Perihelion Distance

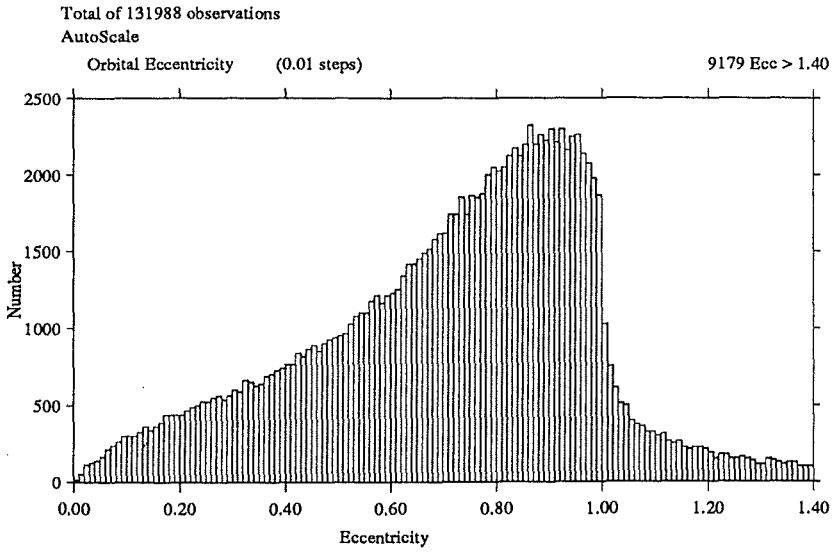


Figure D.2: Orbital Eccentricity

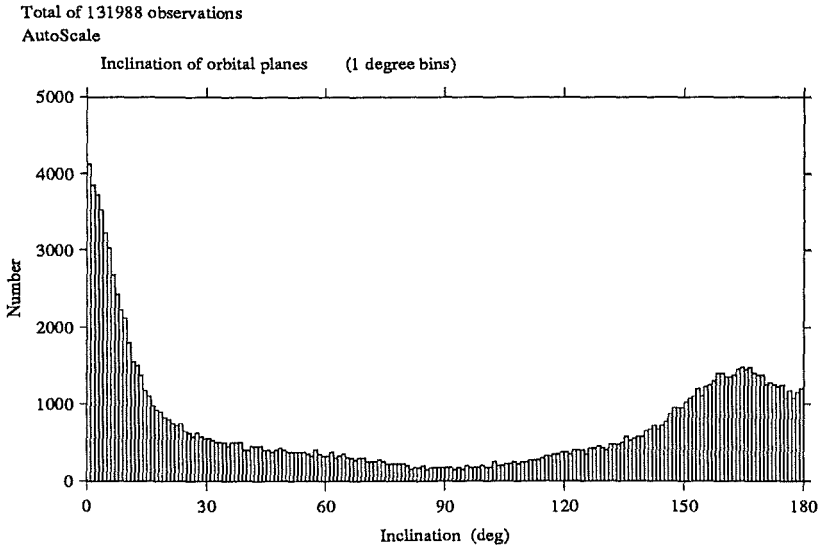


Figure D.3: Inclination of Orbital Plane

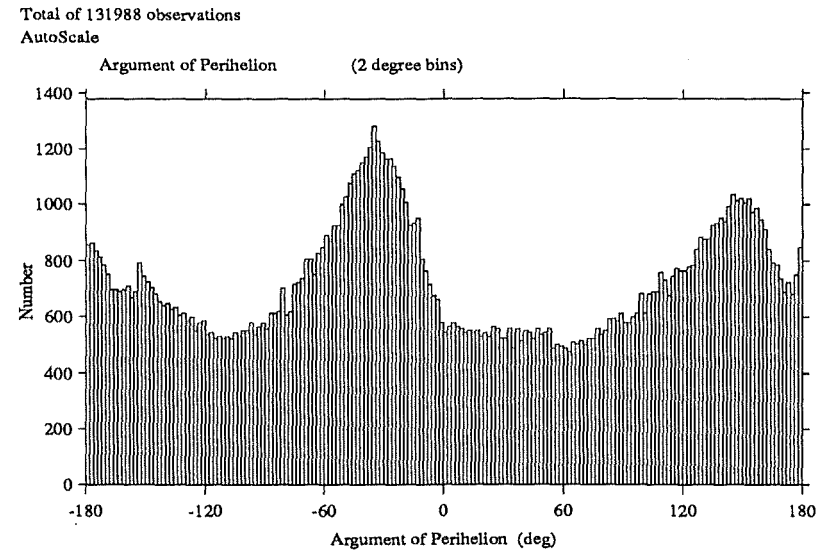


Figure D.4: Argument of Perihelion

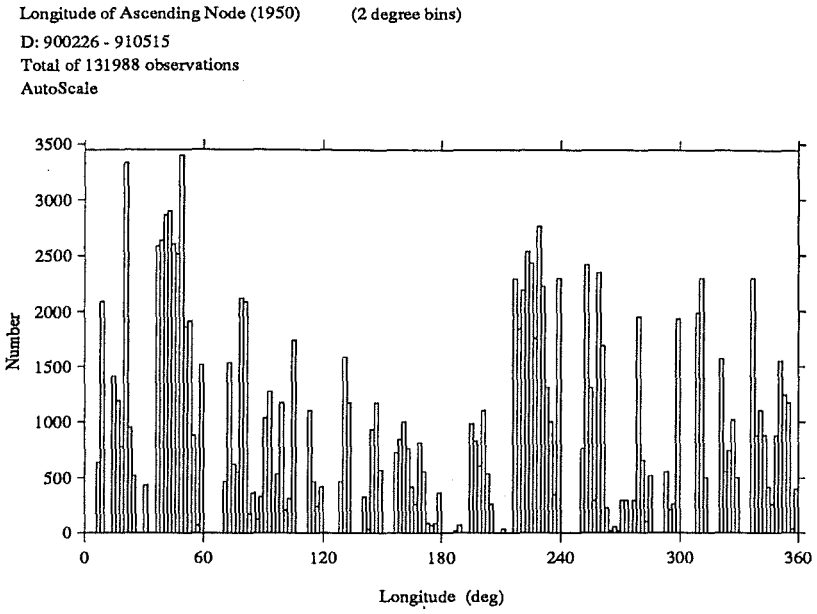


Figure D.5: Longitude of the Ascending Node

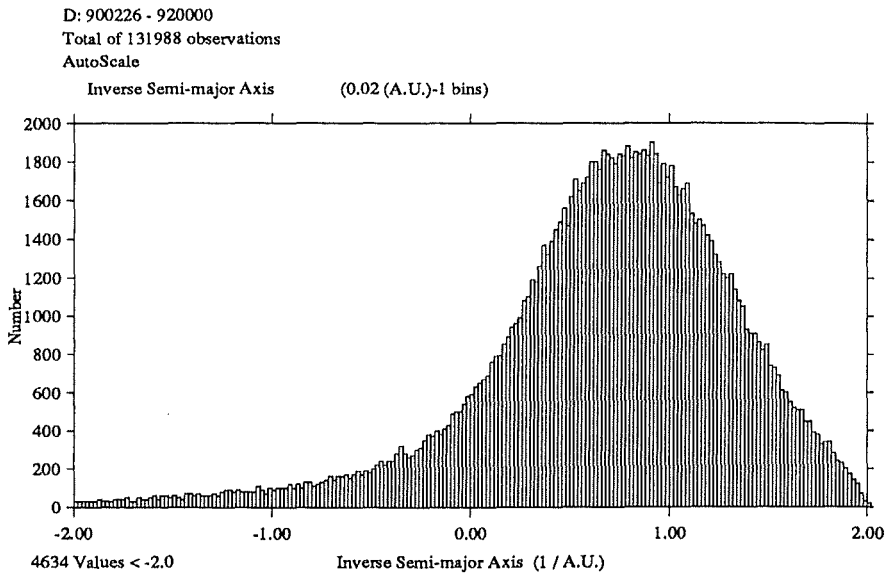


Figure D.6: Inverse Semi-Major Axis

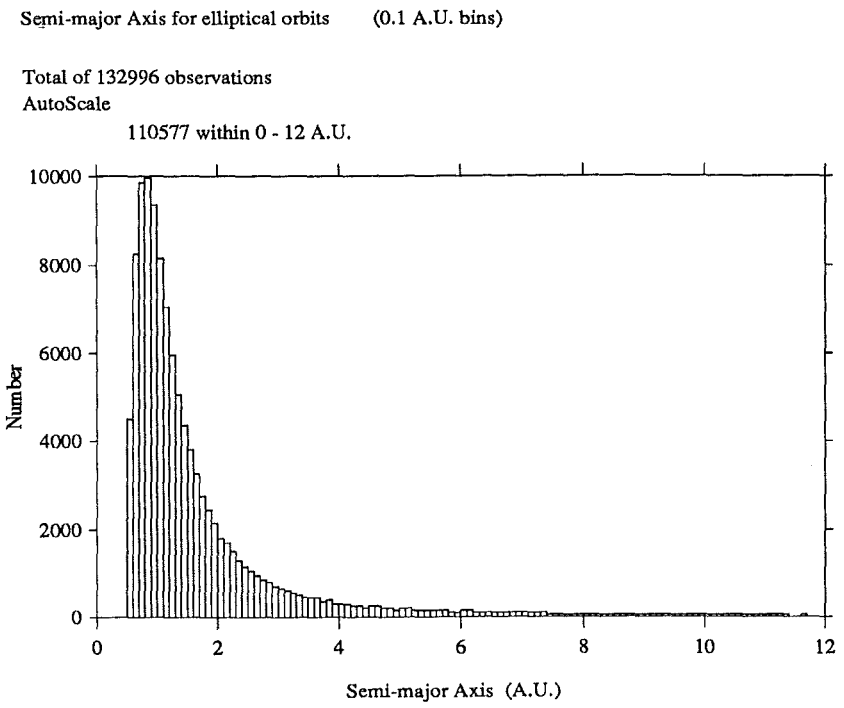


Figure D.7: Semi-Major Axis

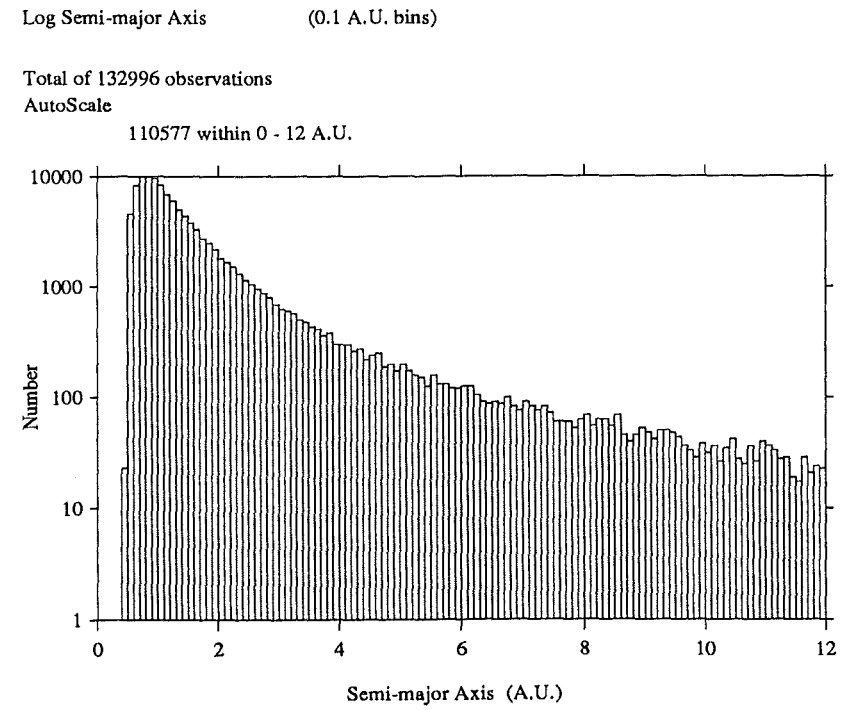


Figure D.8: Log Semi-Major Axis

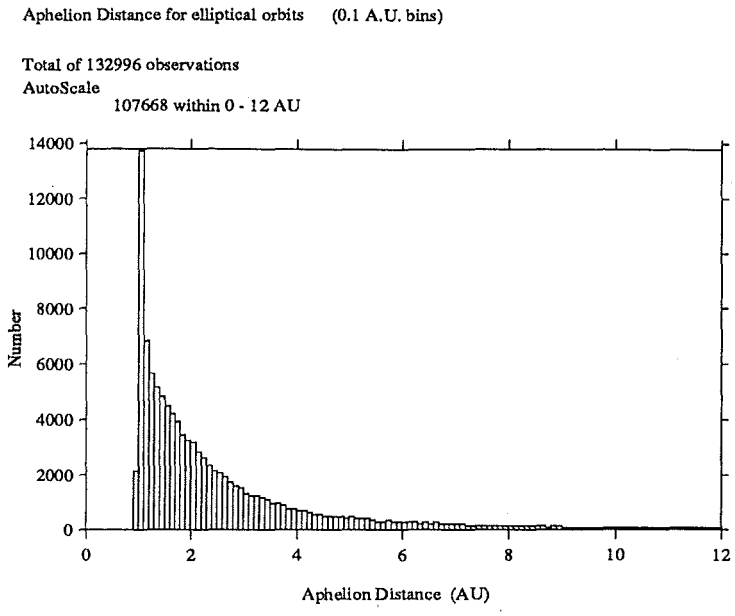


Figure D.9: Aphelion Distance

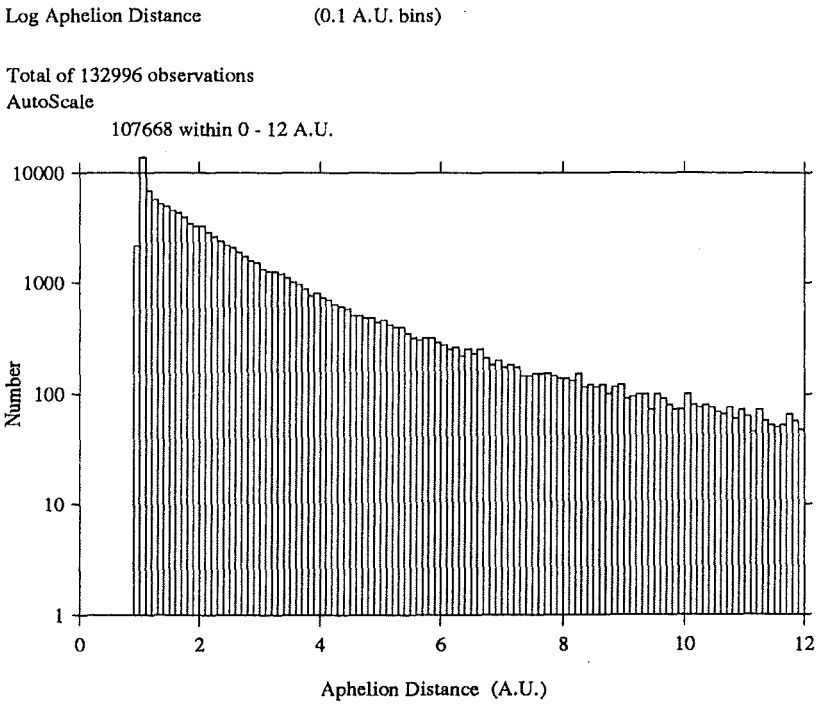


Figure D.10: Log Aphelion Distance

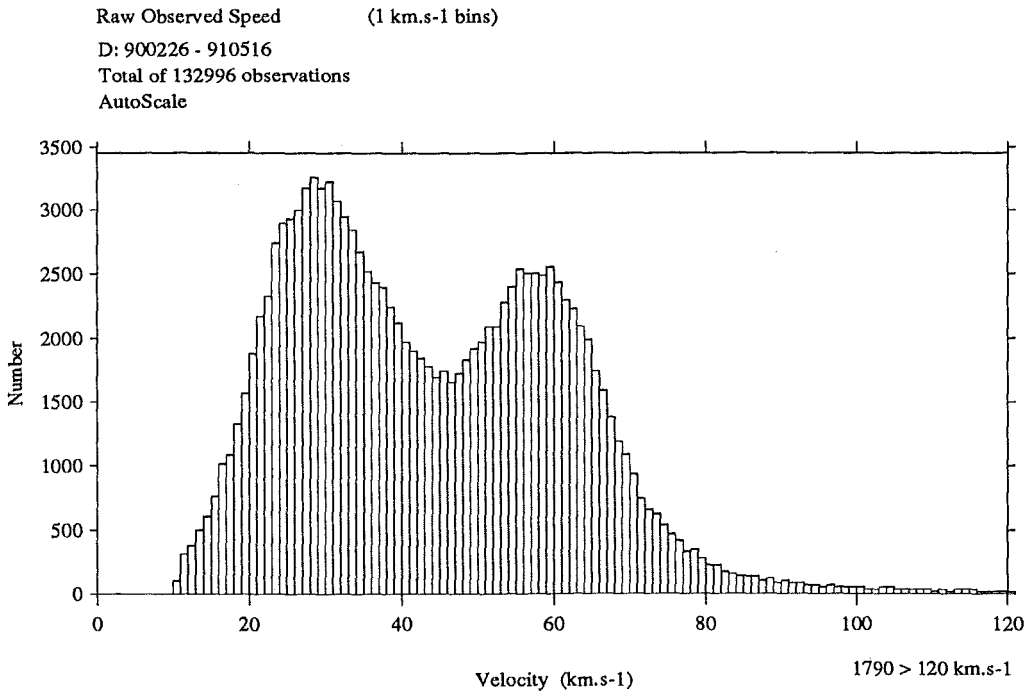


Figure D.11: Observed Atmospheric Speeds

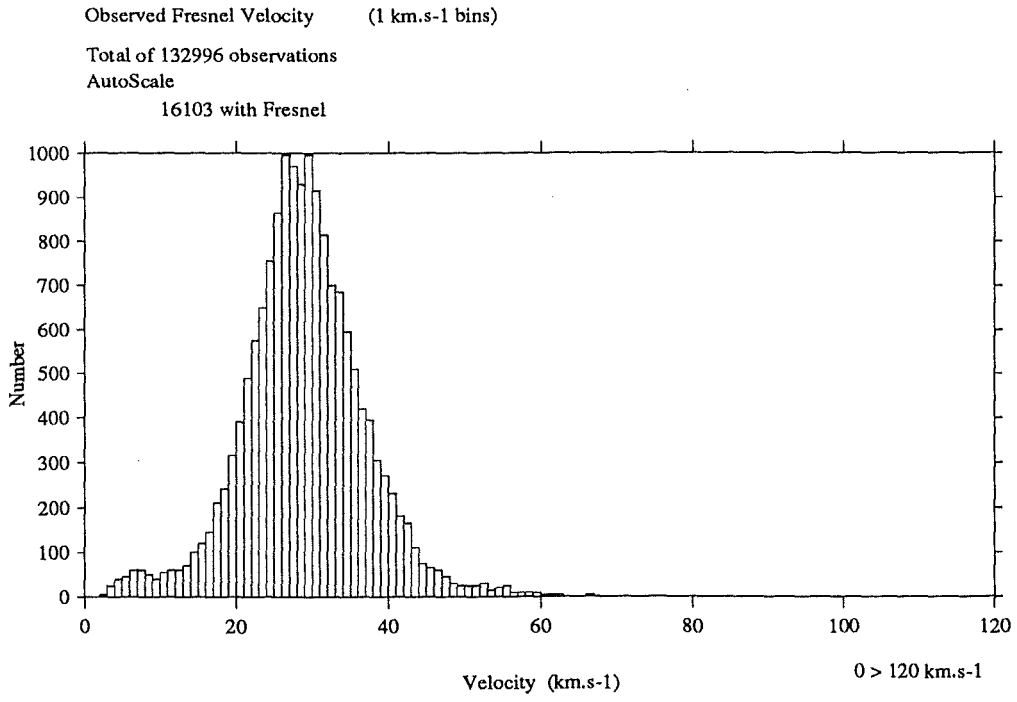


Figure D.12: Fresnel Diffraction Speeds

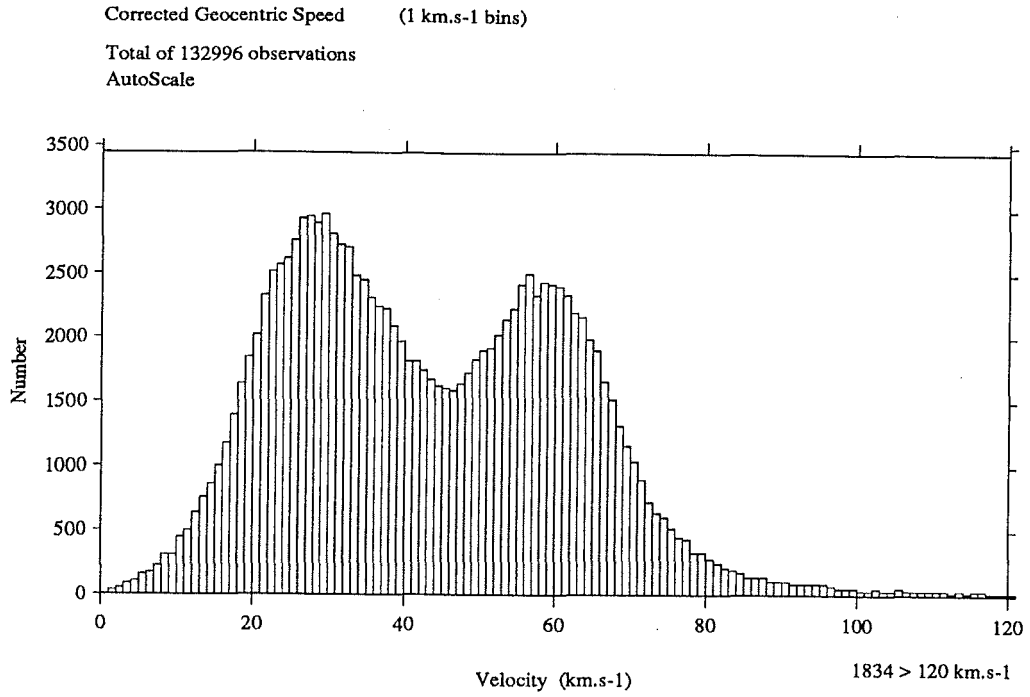


Figure D.13: Corrected Geocentric Speeds

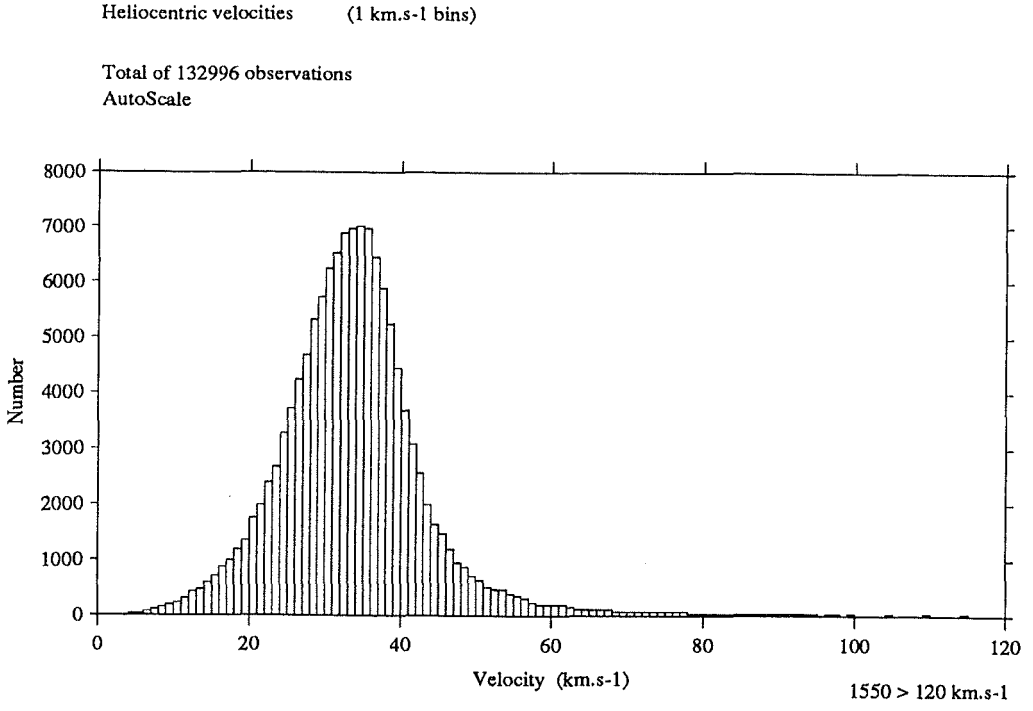


Figure D.14: Heliocentric Speeds

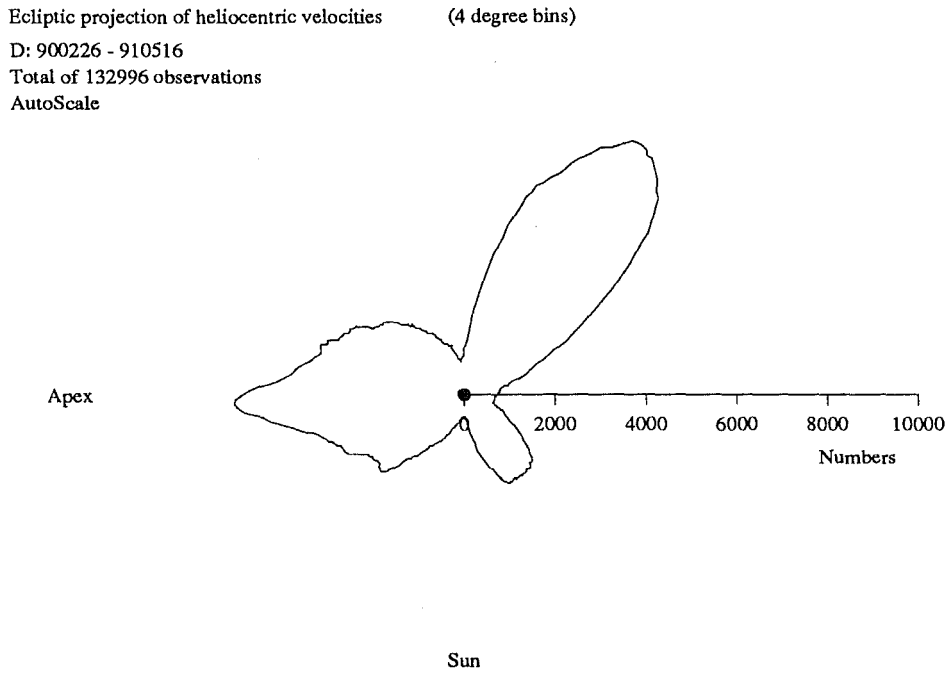


Figure D.15: Ecliptic Projection of Heliocentric Velocity

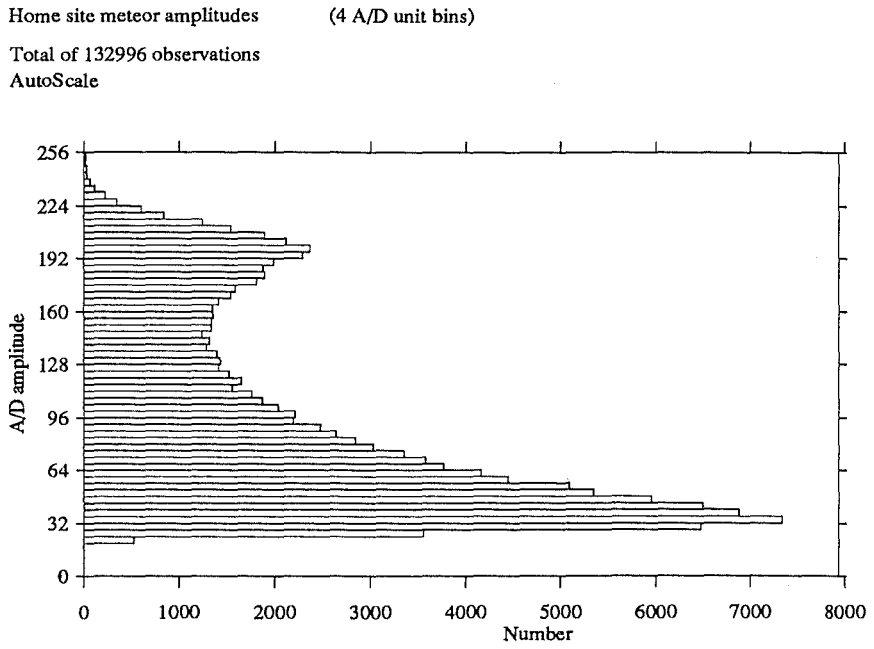


Figure D.16: Home Site Echo Amplitudes

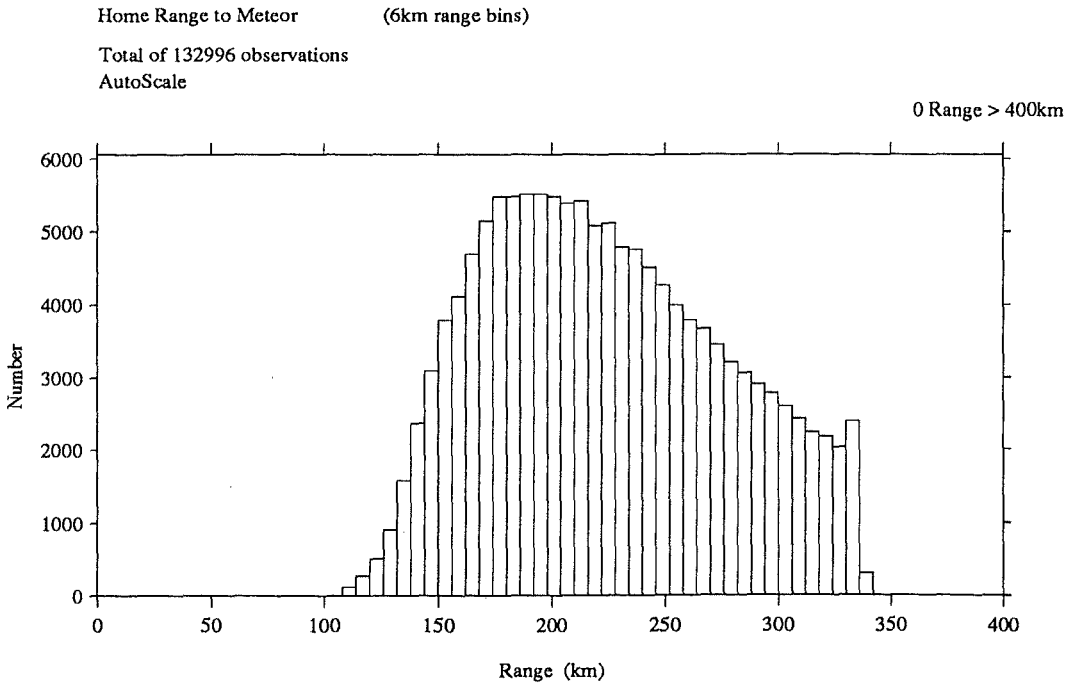


Figure D.17: Home Range to Meteor

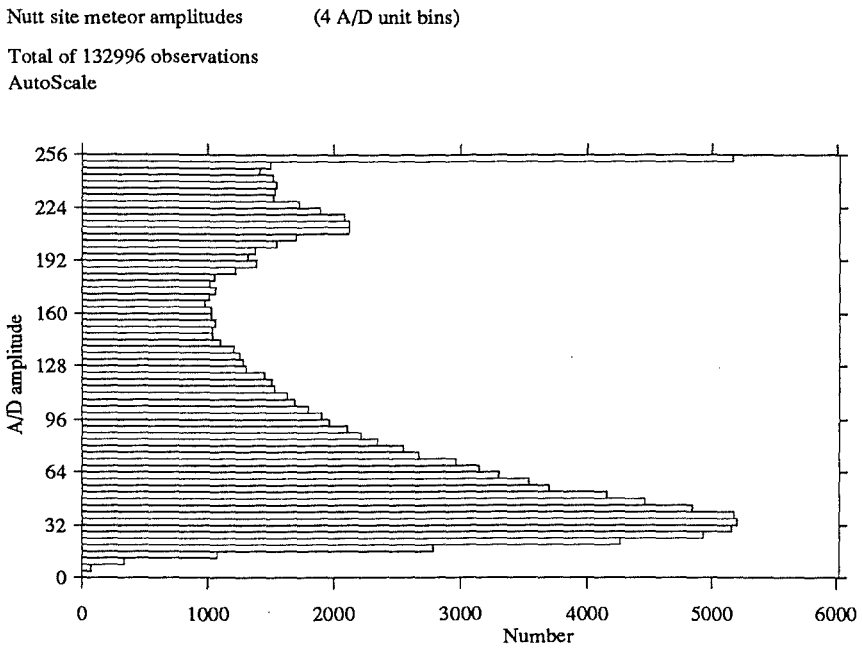


Figure D.18: Nutt Site Echo Amplitudes

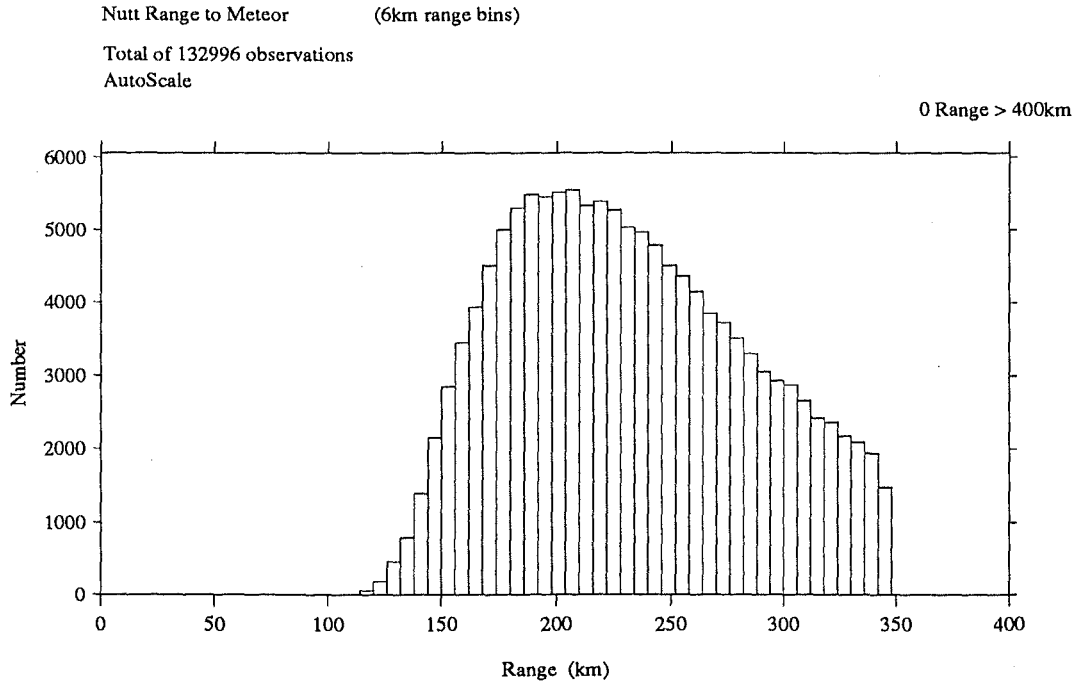


Figure D.19: Nutt Range to Meteor

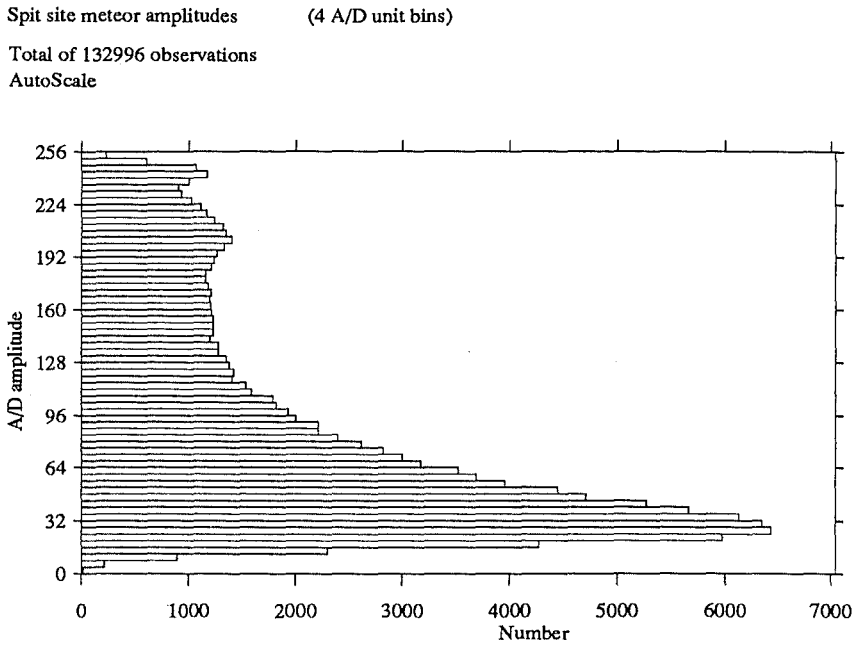


Figure D.20: Spit Site Echo Amplitudes

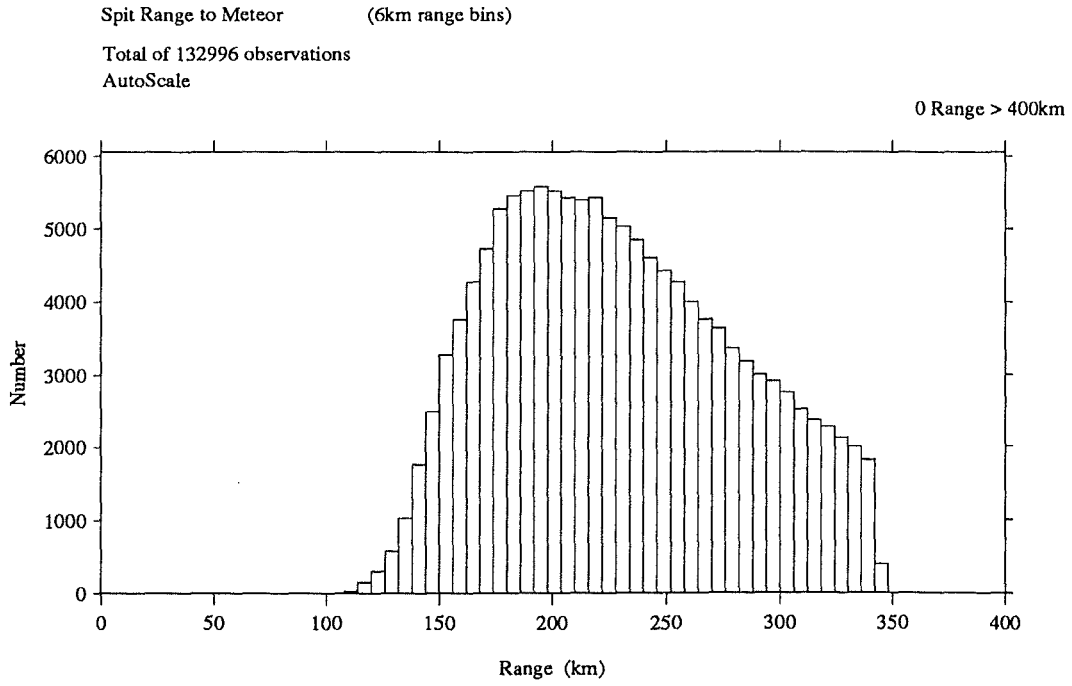


Figure D.21: Spit Range to Meteor

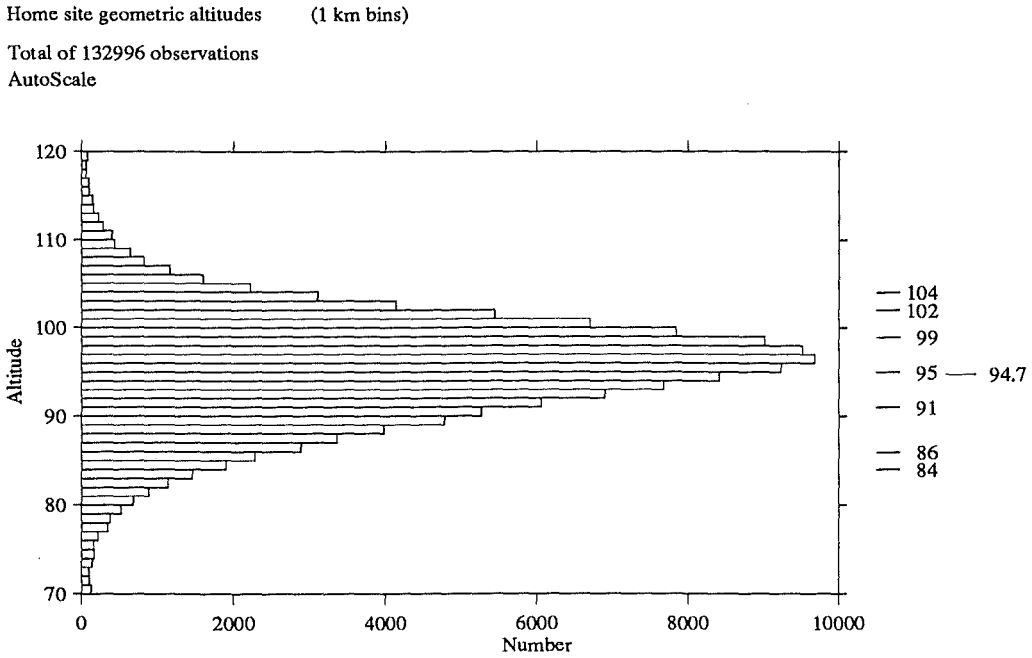


Figure D.22: Geometric Height of Home Site Reflection Point

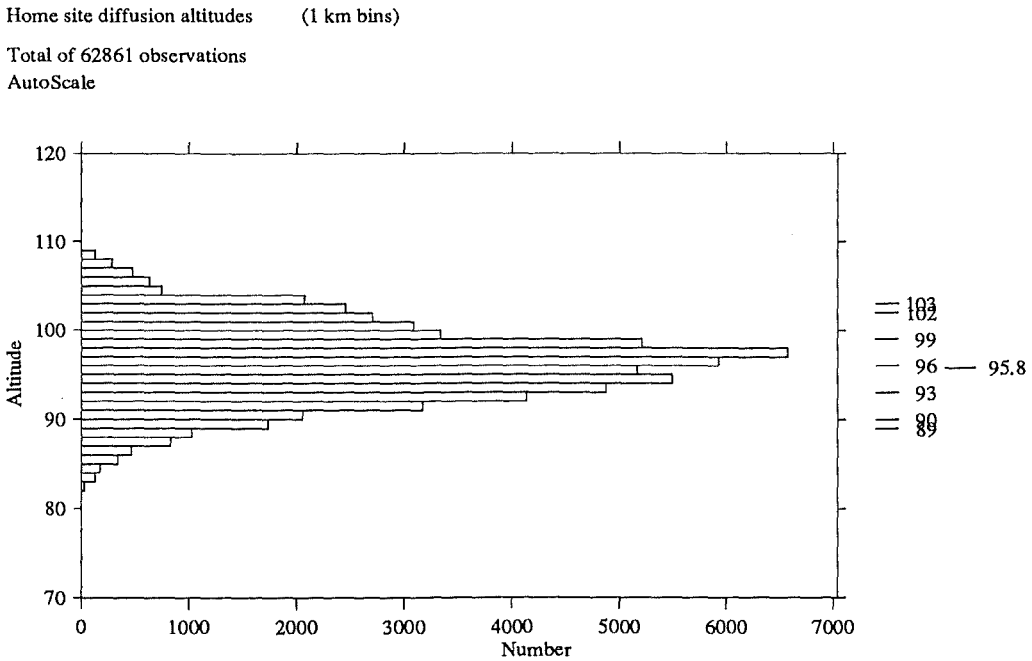


Figure D.23: Diffusion Height of Home Site Reflection Point

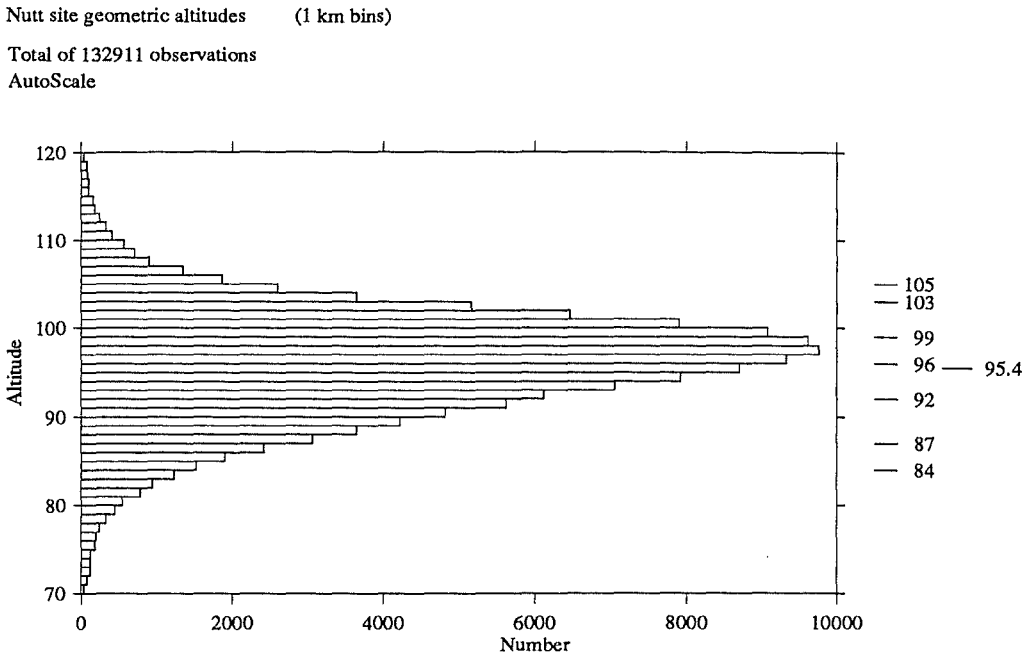


Figure D.24: Geometric Height of Nutt Site Reflection Point

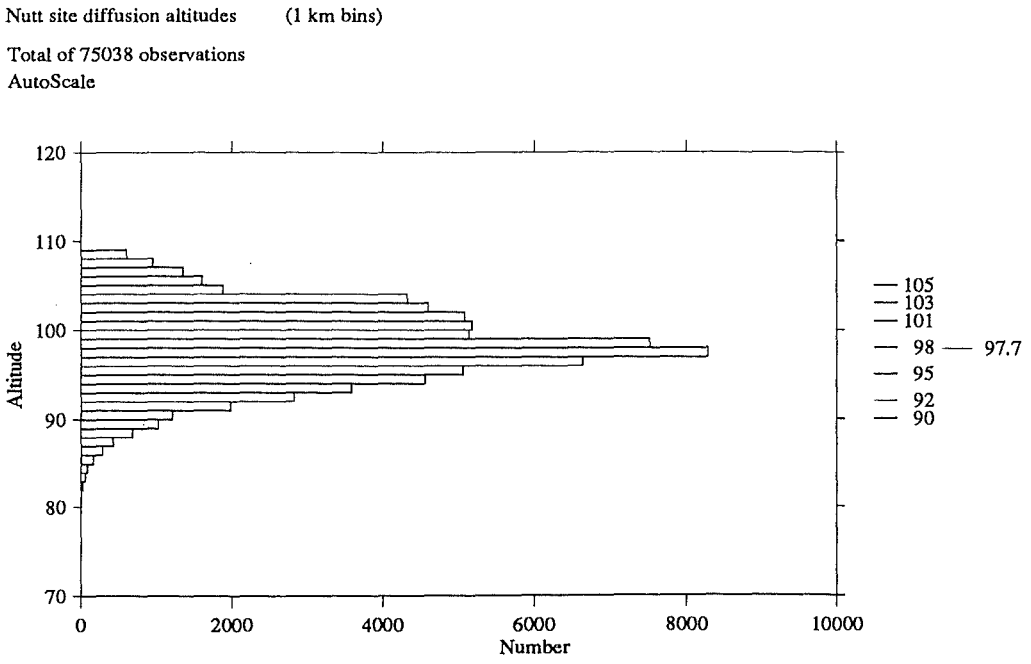


Figure D.25: Diffusion Height of Nutt Site Reflection Point

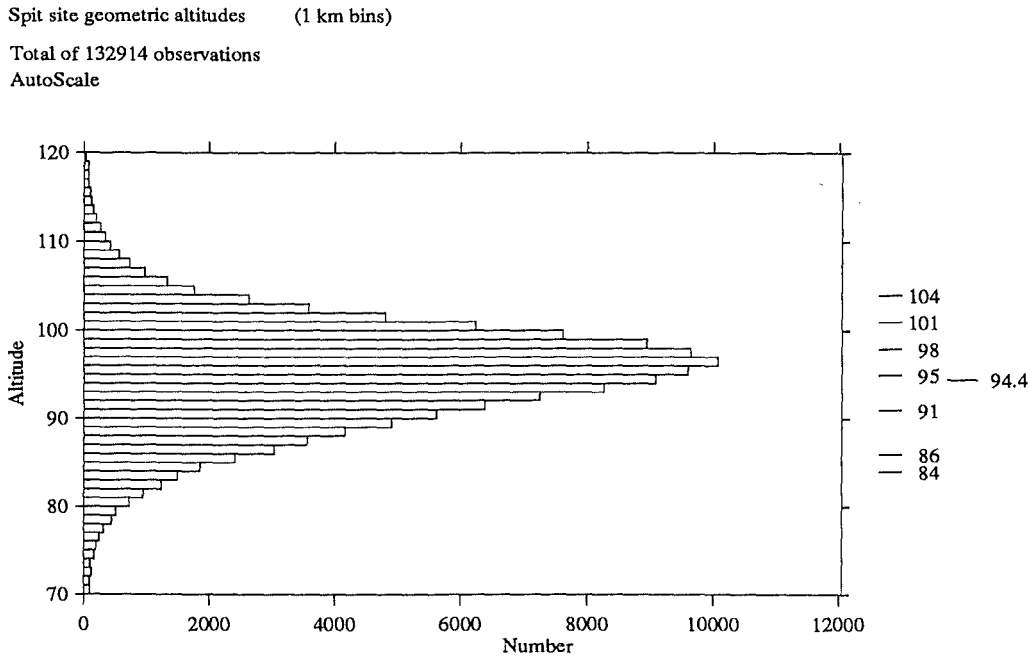


Figure D.26: Geometric Height of Spit Site Reflection Point

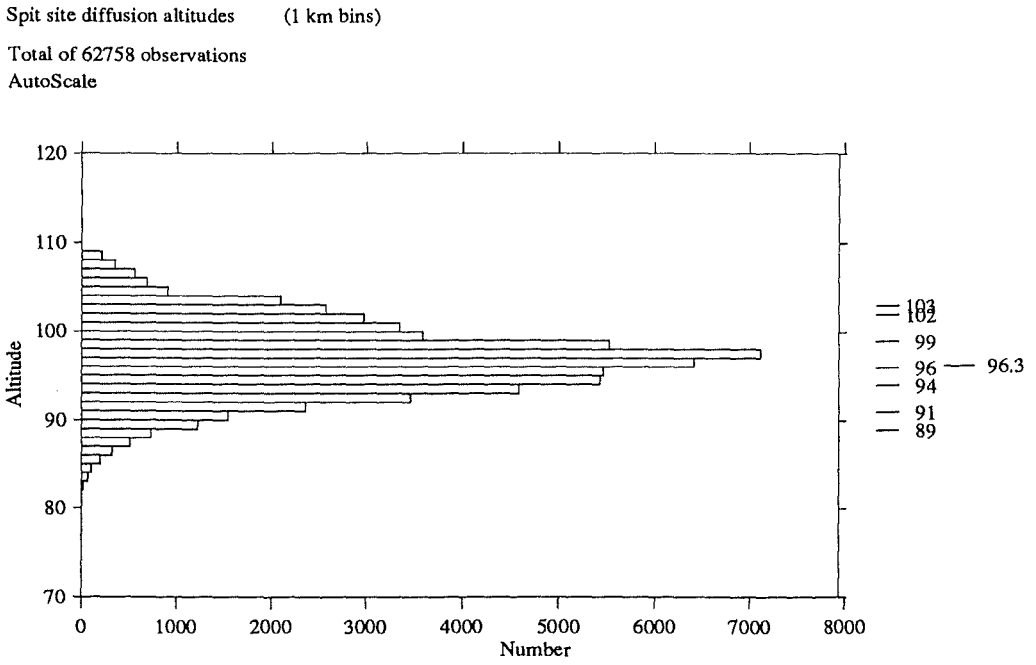


Figure D.27: Diffusion Height of Spit Site Reflection Point

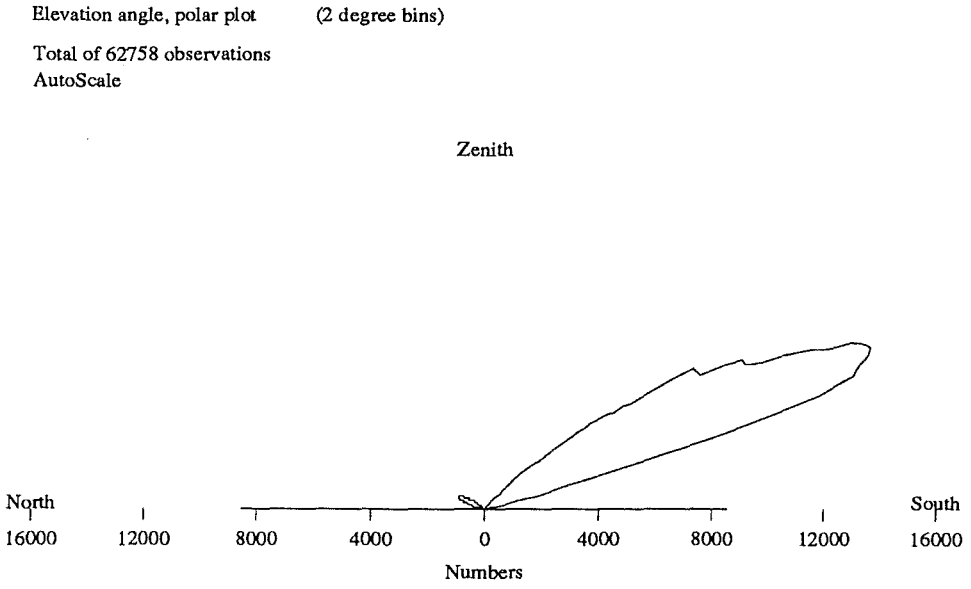


Figure D.28: Elevation Angle, Polar Plot

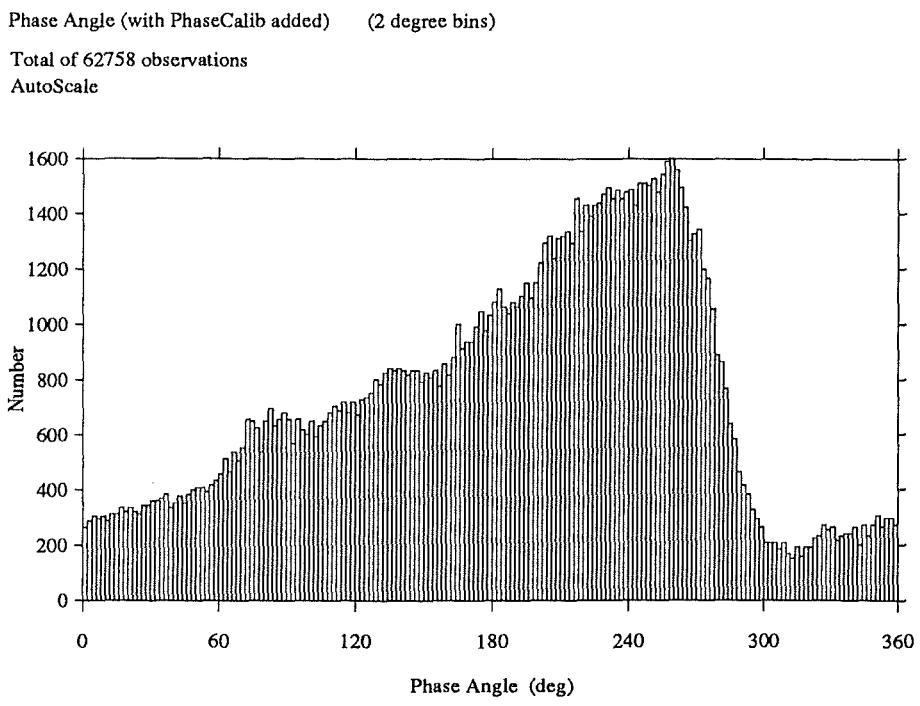


Figure D.29: Relative Home Site Phase Angle

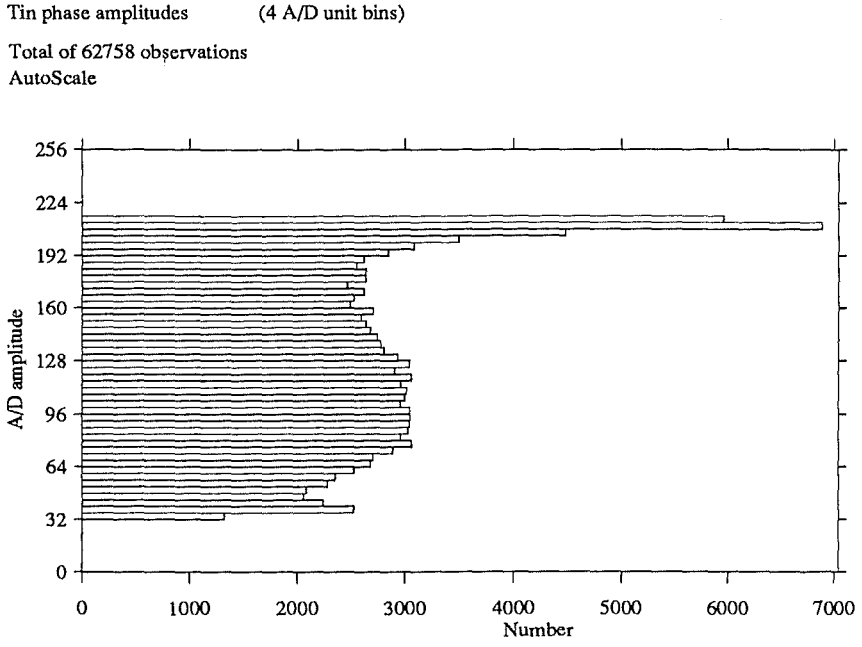


Figure D.30: Tin Phase Amplitudes

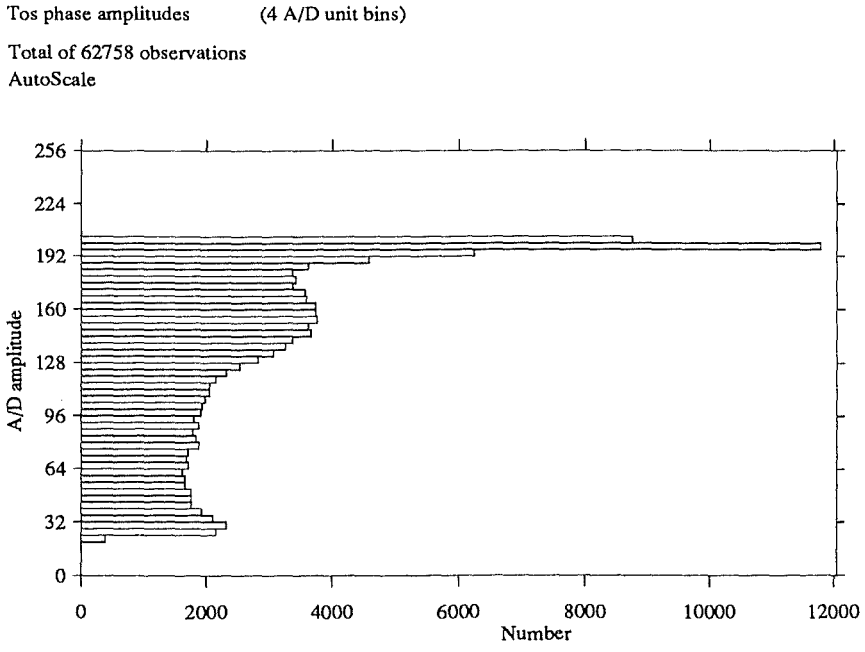


Figure D.31: Tos Phase Amplitudes

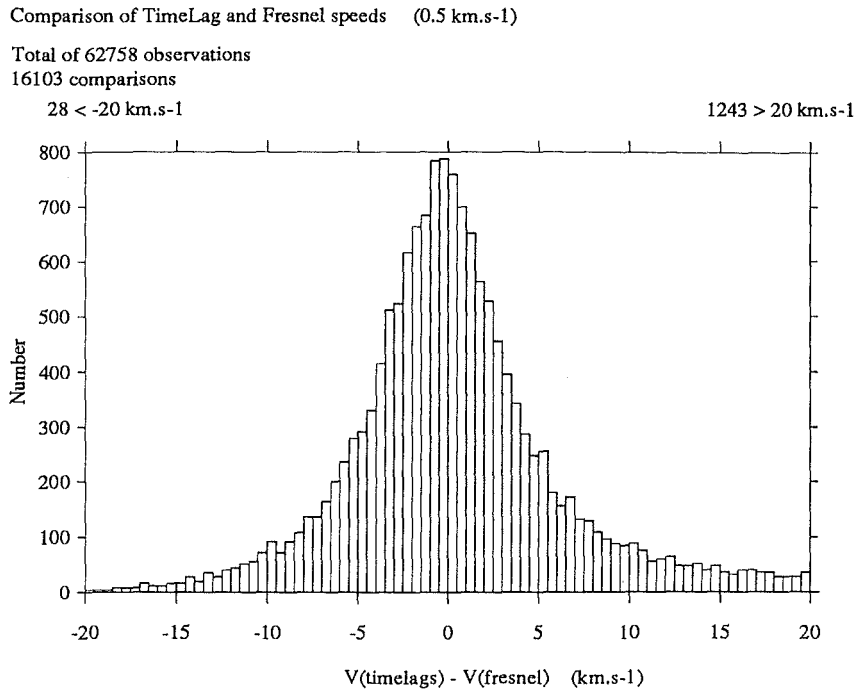


Figure D.32: Comparison of Timelag and Fresnel Speeds

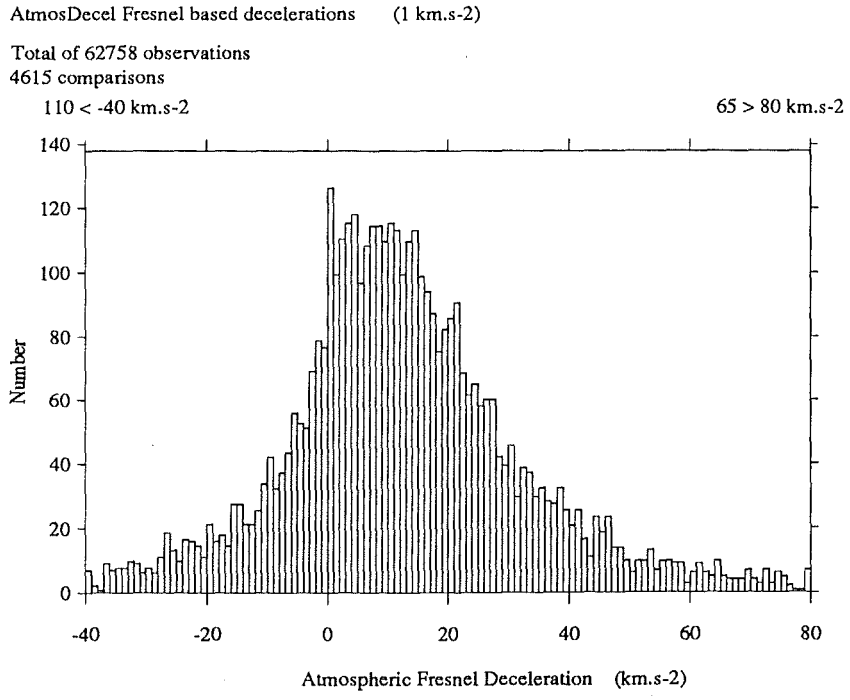


Figure D.33: Fresnel Based Atmospheric Decelerations